

UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO

FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



**SISTEMA DE CONTROL DE PERSONAL Y TAREAJE PARA EL PROYECTO
DE NUTRICIÓN INFANTIL DEL GOBIERNO REGIONAL PUNO - 2015**

TESIS

PRESENTADA POR:

Bach. CANAHUIRE NINA MERY GISSELA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ESTADÍSTICO E INFORMÁTICO

PUNO – PERÚ

2017

UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA
SISTEMA DE CONTROL DE PERSONAL Y TAREAJE PARA EL PROYECTO
DE NUTRICIÓN INFANTIL DEL GOBIERNO REGIONAL PUNO - 2015

TESIS

PRESENTADA POR:

Bach. MERY GISSELA CANAHUIRE NINA

Para optar el Título Profesional de:

INGENIERO ESTADÍSTICO E INFORMÁTICO



APROBADA POR:

PRESIDENTE

:  _____

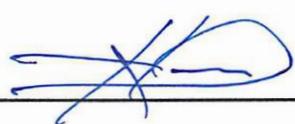
Dr. EDGAR ELOY CARPIO VARGAS

PRIMER MIEMBRO

:  _____

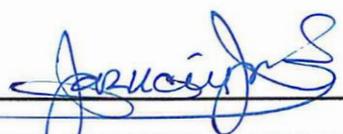
M. Sc. FREDY HERIC VILLASANTE SARAVIA

SEGUNDO MIEMBRO

:  _____

M.Sc. LUIS HUGO HUACASI VASQUEZ

DIRECTOR

:  _____

M.C. CESAR AUGUSTO LLUEN VALLEJOS

ASESOR

:  _____

Mg. EMMA ORFELINDA AZAÑERO DE AGUIRRE

ÁREA : Informática

TEMA : Sistema Informático

FECHA DE SUSTENTACIÓN : 22 de Diciembre de 2017

DEDICATORIA

A dios por ser nuestra guía, a Jesús por ser nuestra inspiración, modelo y por ser el ejemplo más grande de amor en este mundo.

A mis padres: Raul y Mercedes, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar aquí y convertirme en lo que soy. Ha sido un privilegio ser su hija son los mejores padres.

A mi hermanito Kevin por siempre estar a mi lado haciéndome reír en momentos duros y ayudarme cuando siempre lo necesite.

A mi hermosa hija Angely Jeanne, por alegrar mi vida al venir a este mundo, que con su luz ha iluminado mi vida y hace mi camino más claro, eres lo más lindo en mi vida

A mi amado esposo que ha sido el impulso durante toda mi carrera y el pilar principal para la culminación de la misma, que con su apoyo constante y amor incondicional ha sido amigo y compañero inseparable, fuente de sabiduría, calma y consejo en todo momento.

AGRADECIMIENTOS

Agradezco a mi hermosa Universidad Nacional del Altiplano Puno la cual llevo siempre en mi corazón, que me dio todo y abrió sus puertas del conocimiento para mí.

A mi maravillosa Facultad de Ingeniería Estadística e Informática por la Formación Profesional que muchos como yo eligieron esta extraordinaria carrera y que con mucho orgullo, amor, pasión y respeto representare.

Mi agradecimiento y eterna gratitud a los Miembros del Jurado por su brillante participación en la tarea encomendada.

Mi agradecimiento y reconocimiento al Director de Tesis y Asesor de Tesis por su espléndida participación y apoyo constante.

Un agradecimiento y eterna gratificación a los catedráticos de la Facultad de Ingeniería Estadística e Informática por sus conocimientos, consejos, confianza y formación, quienes participaron en mi excelente formación profesional.

A mis compañeros de la Promoción 2011, por la sana competencia y grandes deseos de superación personal, con quienes logramos marcar la diferencia en la Carrera Profesional de Ingeniería Estadística e Informática.

ÍNDICE

RESUMEN.....	10
INTRODUCCIÓN.....	11
ABSTRACT	11
CAPITULO I PLANDE INVESTIGACIÓN.....	14
1.1 PROBLEMA.....	14
1.2 FORMULACIÓN DEL PROBLEMA.....	16
1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN.....	16
1.4 OBJETIVOS	17
1.4.1 Objetivo General:.....	17
1.4.2 Objetivos Específicos.....	18
1.5 HIPÓTESIS DE LA INVESTIGACIÓN.....	18
1.6 LIMITACIONES DE LA INVESTIGACIÓN.....	18
CAPITULO II MARCO TEÓRICO.....	20
2.1 ANTECEDENTES DE LA INVESTIGACIÓN:.....	20
2.2 BASE TEÓRICA	22
2.2.1 INTRODUCCIÓN A LOS SISTEMAS DE INFORMACIÓN.....	22
2.2.2 METODOLOGÍAS DEL DESARROLLO DEL SOFTWARE	27
2.2.3 SISTEMAS GESTORES DE BASES DE DATOS	43
2.2.4 INTRODUCCIÓN AL MICROSOFT SQL SERVER.....	49
2.2.5 VISUAL BASIC	51
2.2.6 INTERFAZ DE USUARIO	54
2.2.7 ISO 9126	56
2.2.8 PRUEBAS O MÉTRICAS DEL SOFTWARE.....	60
2.3 DEFINICIÓN DE TÉRMINOS BÁSICOS.....	67

CAPITULO III MATERIALES Y MÉTODOS	76
3.1 LOCALIZACIÓN	76
3.2 POBLACIÓN.....	76
3.3 MÉTODO DE RECOLECCIÓN DE LA INFORMACIÓN.....	77
3.4 METODOLOGÍA PARA EL DESARROLLO DEL SISTEMA.....	78
3.4.1 MÉTODO DEL CICLO DE VIDA PARA EL DESARROLLO DE SISTEMAS	78
3.5 MATERIAL EXPERIMENTAL	83
3.6 REQUERIMIENTOS MÍNIMOS DEL SISTEMA	84
CAPITULO IV RESULTADOS Y DISCUSIONES	85
4.1 ANÁLISIS DEL SISTEMA	85
4.1.1 ANÁLISIS DE LOS REQUISITOS DE SOFTWARE	86
4.1.2 DISEÑO.....	92
4.1.3 REQUERIMIENTOS FUNCIONALES	94
4.1.4 REQUERIMIENTOS NO FUNCIONALES	95
4.1.5 GENERACIÓN DE CÓDIGO.....	96
4.1.6 PRUEBAS E IMPLANTACIÓN	97
4.2 DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA.....	98
4.2.1 Pantalla de acceso al sistema.....	98
4.2.2 Ventana principal del sistema	99
4.2.3 Modulo de Mantenimiento.....	99
4.2.4 Módulo registro de Personal	100
4.2.5 Módulo de Operaciones.....	100
4.2.6 Módulo de Consultas	100
4.2.7 Módulo de Reportes.....	101
4.2.8 Módulo de Base de Datos.....	101

4.2.9 Módulo de Configuración	101
4.3 PARÁMETROS DE PRUEBA DEL SISTEMA.....	102
4.3.1 Prueba de Caja Blanca	102
4.3.2 Métrica de punto función.....	103
4.3.3 PRUEBA DE SOFTWARE ESTÁNDAR ISO 9126.....	105
CONCLUSIONES.....	110
RECOMENDACIONES Y SUGERENCIAS	111
BIBLIOGRAFÍA.....	112
ANEXOS	115

ÍNDICE DE TABLAS

Tabla 1: Costos generales	91
Tabla 2: Valores del dominio del sistema de personal y tareaje	103
Tabla 3: Valores de ajuste de la complejidad del Sistema.....	104
Tabla 4: Resultados de la evaluación de la calidad del producto de software del Sistema de Personal y Tareaje según el estándar ISO – 9126.....	106
Tabla 5: Tabla de valores de puntuación para la evaluación de la calidad del producto de software estándar ISO – 9126.....	108
Tabla 6: Resultado de la Validación de la Calidad del Producto de Software del Sistema	108

ÍNDICE DE FIGURAS

Figura N° 1: Modelo General de un Sistema.....	23
Figura N° 2: Tipología de usuarios de sistemas.....	25
Figura N° 3: El modelo lineal secuencial.....	29
Figura N° 4: Modelo en espiral de Boehm para el proceso del software (IEEE, 1998).....	32
Figura N° 5: Un modelo espiral típico.....	34
Figura N° 6: El modelo de construcción de prototipos.....	36
Figura N° 7: Pantalla de una base de datos.....	44
Figura N° 8: Ciclo de Vida Clásico de los Sistemas.....	79
Figura N° 9: Diagrama de Contexto del Sistema.....	86
Figura N° 10: Diagrama de Casos de Uso Sistema de Pernal y Tareaje.....	87
Figura N° 11: Diagrama de clase de uso - Administración de Usuarios.....	87
Figura N° 12: Diagrama de casos de uso - Reportes del Sistema.....	88
Figura N° 13: Diagrama de clases del sistema de personal y tareaje.....	89
Figura N° 14: Diagrama del Modelo Entidad - Relación.....	93
Figura N° 15: Diagrama del Modelo Relacional.....	94
Figura N° 16: Ventana de Acceso al Sistema.....	98
Figura N° 17: Ventana Principal del Sistema.....	99
Figura N° 18: Módulo de Mantenimiento del Sistema.....	99
Figura N° 19: Modulo de Registro de Personal.....	100
Figura N° 20: Modulo de Operaciones.....	100
Figura N° 21: Modulo de Consultas.....	100
Figura N° 22: Modulo de Reportes.....	101
Figura N° 23: Modulo de Base Datos.....	101
Figura N° 24: Modulo de Configuración.....	101

RESUMEN

La oficina de administración del Proyecto de Nutrición Infantil del Gobierno Regional Puno, controla asistencias, inasistencias, tardanzas, permisos, tareajes, etc. Las cuales se gestionan de forma manual, por tanto esta en búsqueda de solucionar el problema de la desactualización y manejo de datos para un efectivo control del personal. La presente tesis tiene por objetivo; desarrollar un Sistema de Control de Personal y Tareaje eficiente que agilice y dinamice el trabajo administrativo requerido en el Proyecto. Para el desarrollo del sistema se utilizó la metodología en cascada, ya que a través de cada una de sus fases de análisis, diseño, codificación, pruebas, entrega, y mantenimiento fue posible lograr cumplir los requerimientos y necesidades que el Proyecto demandaba. Para el diseño se utilizó el modelamiento basado en UML, se midió la calidad del software aplicando la ficha de evaluación ISO 9126 a los administradores del sistema. Así mismo el gestor de base de datos SQL Server para el almacenado y modelamiento de datos. Concluyendo; Que con el desarrollo e implementación del sistema de control de personal y tareaje se agilizo y dinamizo eficientemente los procesos administrativos en el Proyecto de Nutrición Infantil del Gobierno Regional Puno, obteniendo de esta manera las asistencias del personal colaborador y sus respectivo tareaje para la posterior elaboración de la planilla de remuneraciones y pago. De acuerdo a la ficha de evaluación ISO- 9126 se demostró que el sistema **cumple con los requisitos** en cuanto se refiere a la calidad del producto del software.

Palabras Clave: Sistema de Control Personal, Metodología en Cascada, Tareaje, Eficiencia.

ABSTRACT

The administration office of the Child Nutrition Project of the Puno Regional Government, controls attendance, absences, delays, permits, homework, etc. Which are managed manually, so is looking to solve the problem of outdated and data management for effective control of staff. The present thesis is aimed at; develop an efficient Personnel Control and Talent Control System that streamlines and dynamizes the administrative work required in the Project. For the development of the system, the cascade methodology was used, since through each of its phases of analysis, design, coding, testing, delivery, and maintenance it was possible to achieve the requirements and needs that the Project demanded. UML-based modeling was used for the design, the quality of the software was measured by applying the ISO 9126 evaluation form to the system administrators. Also the SQL Server database manager for the storage and modeling of data. Concluding; That with the development and implementation of the personnel and task control system, the administrative processes in the Child Nutrition Project of the Puno Regional Government were efficiently streamlined and improved, thus obtaining the assistance of the collaborating personnel and their respective tasks for the subsequent elaboration of the remuneration and payment form. According to the ISO-9126 evaluation sheet, it was shown that the system **meets the requirements** as regards the quality of the software product.

Key words: Personal Control System, Cascade Methodology, Tareaje, Efficiency.

INTRODUCCIÓN

Actualmente la tecnología constituye un elemento de esencial interés en todos los sectores de la sociedad; entre otras razones, por la facilidad y comodidad que le ha brindado al hombre para sobrevivir. La tecnología de información día a día se mejora para incorporar nuevos ámbitos de la sociedad a la misma. En paralelo encontramos los sistemas de información, los cuales agilizan la gestión de los procesos dentro de las organizaciones, en este sentido se ejecutan procesos largos, tediosos y repetitivos en menor tiempo y con mayor precisión. A este proceso se ha unido organizaciones de todo tipo, tales como gubernamentales, bancarias y comerciales.

En el Proyecto de Nutrición Infantil del Gobierno Regional de Puno, a fin de agilizar sus procesos administrativos internos, tales como el control de asistencia personal, tareaje y el reporte, actualmente existen procesos que no funcionan apropiadamente, por tal motivo deben ser adaptados a la nueva tecnología de la información para ofrecer mayores y mejores servicios a su personal, quienes actualmente no poseen un sistema de información automatizado.

Es por esto que en la presente investigación se incluyen todos los elementos que interactúan con el sistema creado, además de la fundamentación, la cual se presenta en cuatro capítulos a continuación:

CAPÍTULO I: Expone el planteamiento del problema, donde se revisan los antecedentes, objetivos, los alcances y límites en el desarrollo del sistema.

CAPÍTULO II: Se explican los fundamentos teóricos involucrados en el desarrollo del sistema de control y tareaje.

CAPÍTULO III: Se muestra las especificaciones de los equipos utilizados conjuntamente con la metodología empleada, explicando cada una de las faces involucradas en el proyecto.

CAPÍTULO IV: Se da a conocer los resultados y discusión, la implementación y pruebas realizadas, señalar que se encuentra especificado el procedimiento que se llevó a cabo durante el desarrollo de dicho sistema, mostrando desde de la identificación del problema que la institución presenta, planeación, hipótesis y justificación. Seguidamente se procedió a describir las conclusiones y recomendaciones obtenidas durante el proceso de desarrollo del sistema, tomando en cuenta las observaciones encontradas durante el proceso. También, se hace mención a las referencias bibliográficas consultadas referentes a metodología de la investigación y al tema objeto de investigación.

CAPITULO I

PLAN DE INVESTIGACIÓN

1.1 PROBLEMA

Una empresa y/o institución de todos los tamaños requiere contar con herramientas de gestión para controlar y cuantificar los distintos aspectos de su operación. El software de aplicación para control de personal, almacén, planillas, ventas, entre muchos otros son usuales en la gran mayoría de empresas.

Una de las áreas que más ha tardado en automatizarse ha sido la del control de asistencia o permanencia y tareaje del personal, debido en buena cuenta por la aun amplia utilización de sistemas tradicionales, llámese “sistemas manuales”, de control basados en los antiguos relojes mecánicos que utilizan tarjetas de cartón, o únicamente con “partes de asistencia” que no son otra cosa que hojas de papel con líneas numeradas

en las que los empleados firman e indican, ellos mismos, la hora en la cual ingresan o salen del trabajo .

Los sistemas de control de asistencia modernos se basan en tecnologías, pues estando en la era de la tecnología es algo que no debería pasar. Otros de los propósitos de esta investigación es, automatizar el proceso de control del personal y tareaje, Verificar el funcionamiento del sistema y determinar la reducción de tiempo en las labores administrativas durante el periodo 2015.

El tiempo en las instituciones y en todo sector empresarial es muy importante, porque es un recurso que significa dinero, y en muchas empresas los problemas más resaltantes son las tardanzas e inasistencias en las mismas.

Por lo mismo que no cuentan con un sistema para el control del personal que labora en las instituciones, ya que muchas de ellas utilizan el control de forma manual, pero que trae malos resultados, pues estando en la era de la tecnología es algo que no debería pasar.

Ya que al no tener el control del personal que labora, los mismos muchas veces se exceden en las continuas tardanzas o en las inasistencias, y así no pueden afrontar las actividades de su vida laboral adecuadamente debido al menor tiempo empleado ocasionado por la impuntualidad o la ausencia y al mismo tiempo las instituciones del estado hacen gastos excesivos, al pagar al personal por horas que no han sido trabajadas.

La realización del registro del personal y tareaje es efectuado manualmente, lo cual provoca inexactitud al llevar a cabo la remuneración económica de los colaboradores, entre ellos, excedente en la nómina y a su vez, menor pago en las actividades u horarios que laboran en la institución

Con la necesidad de agilizar y dinamizar el trabajo administrativo dentro del Proyecto de Nutrición Infantil, lleva pues a sus administradores a manifestar que hay Carencia de un Sistema de control de personal y tareaje.

1.2 FORMULACIÓN DEL PROBLEMA

¿Mejorara eficientemente el proceso de control de personal implementando el Sistema de Control de Personal y Tareaje en el Proyecto de Nutrición Infantil del Gobierno Regional Puno?

1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN

La presente investigación surge de la necesidad de que el Proyecto de Nutrición Infantil del Gobierno Regional Puno, no dispone de un sistema de control de asistencia y tareaje automatizado, por lo que se pretende realizar un software que permita facilitar las consultas de la información, agilizar los procesos para contribuir al funcionamiento eficiente y eficaz de las operaciones, siendo esta confiable, oportuna, precisa y actualizada.

Es de vital importancia la creación de un mecanismo que ayude a garantizar tanto seguridad como rapidez la administración de la información con el fin

de brindar un mejor servicio las faltas de asistencias por parte de los trabajadores.

Dado los limitados recursos que brinda el gobierno hacia este proyecto, y la necesidad que los trabajadores cumplan con su asistencia en la impartición de sus funciones se requiere de un método eficaz y productivo el cual venga a diseminar el problema del incumplimiento del horario de trabajo en el proyecto.

Es de vital importancia el apoyo que brinda la oficina de administración a la coordinación general, y para ello se apoyará a un proceso que entre otras realizan la actividad de control de personal y tareaje de los trabajadores.

Este sistema será de gran ayuda a la organización administrativa ya que permitirá manejar de manera eficiente las asistencias y tareajes del personal por día, semana, quincena y meses, de una forma eficiente ya que la institución requiere este tipo de organización lo cual sería de vital importancia para la toma de decisiones para el logro de los objetivos propuestos.

1.4 OBJETIVOS

1.4.1 Objetivo General:

Desarrollar e implementar un Sistema de control de personal y tareaje eficiente para el Proyecto de Nutrición Infantil del Gobierno Regional Puno.

1.4.2 Objetivos Específicos

- ✓ Identificar los requerimientos para el sistema de control de personal y tareaje en base a metodología en cascada.
- ✓ Diseñar la estructura física y lógica de la base de datos e interfaz gráfica del sistema mediante la diagramación UML.
- ✓ Realizar la prueba de la calidad de software mediante el ISO 9126, para la validación del sistema.
- ✓ Implementar el sistema en el Proyecto de Nutrición Infantil del Gobierno Regional Puno.

1.5 HIPÓTESIS DE LA INVESTIGACIÓN

La implementación del Sistema para el control de personal y tareaje contribuirá eficientemente a la organización profesional en los procesos administrativos en el proyecto de Nutrición Infantil del Gobierno Regional Puno.

1.6 LIMITACIONES DE LA INVESTIGACIÓN.

Se han definido los límites de la automatización del proceso de asistencia para contener claramente que problema se analizará de la mejor manera y que aspectos se incluirán en la realización de la solución.

- ✓ Se enfoca en la actividad de ingreso, control y consultas de información de las asistencias de trabajadores que realiza la oficina de administración: el proyecto puede aplicarse a diferentes campos, pero por el sentido social del proyecto se desarrollará en el área de administración en la asistencia de trabajadores para un constante control de personal.

- ✓ No se incluirá la generación de datos de asistencia de los trabajadores por ser parte de la entrada al proceso del manejo y control de la información de presencia del personal el cual se quiere solucionar.

La implementación de un sistema como este, conlleva diferentes cambios y fases para asegurarse que este sea el adecuado, tales como:

1. Trabajo más inteligente.
2. Cambio global en el concepto de integración de actividades administrativas de la institución.
3. Las personas que trabajan con el sistema de control del personal dominan la fuerza de trabajo.

CAPITULO II

MARCO TEÓRICO

2.1 ANTECEDENTES DE LA INVESTIGACIÓN:

López, C. (2008). En su investigación refiere que el actual sistema de control de asistencia basado en el mercado de tarjetas es un sistema vulnerable, el modo en el que controla la asistencia es ineficiente, puesto que realiza en control de tarjetas plásticas y no de los trabajadores en sí.

Con la implantación del sistema biométrico de control de asistencia se demostró que existe una alta tasa de suplantación en el mercado de tarjetas, esta suplantación evita el logro de los objetivos trazados por todas las áreas funcionales de la Universidad Nacional de San Martín, tengamos en cuenta que la Universidad tiene alta influencia en el desarrollo de nuestra localidad y que la imagen que proyecta está directamente ligada al comportamiento de su personal.

Juárez, G. (2011). Los tesisistas han podido simular el desarrollo de un software de control y registró con la tecnología biométrica basados en huella digital con la ayuda de un scanner de papel, el software que sirvió para procesamiento y con la ayuda de base de datos para su implementación. También ha analizado, la implementación y fue comparado algunos de los algoritmos que existen para el procesamiento de las huellas dactilares, la implementación del Software se realizó con la ayuda de Matlab y su módulo de desarrollo, el cual permitió realizar el procesamiento de imágenes y su interfaz de usuario.

Campos, W. (2010). Los tesisistas han visto que la aplicación provee de consultas que ayudan a la toma de decisiones por parte de los encargados de la asistencia docente, por lo tanto implicará el seguimiento necesario al ser evaluado.

El código fuente del sistema, su documentación interna y externa fue concedido a Centro de Cálculo para brindar un seguimiento continuo a la aplicación.

Alvarado, J. (2010). En su investigación refiere que el sistema de control administrativo de personal también identificado por sus siglas SICAP, forma una interfaz entre la gestión de recursos humanos y la tecnología de información. Es decir, combina los recursos humanos y en particular sus actividades administrativas con los medios puestos a su disposición por la informática, y se refieren en particular a las actividades de planificación y tratamiento de datos para integrarlos en un único sistema de gestión.

El SICAP contiene un nuevo potencial tecnológico, de innovar. Esto involucra un cambio cultural, que amerita una intervención de Desarrollo Organizacional para garantizar su éxito.

2.2 BASE TEÓRICA

2.2.1 INTRODUCCIÓN A LOS SISTEMAS DE INFORMACIÓN

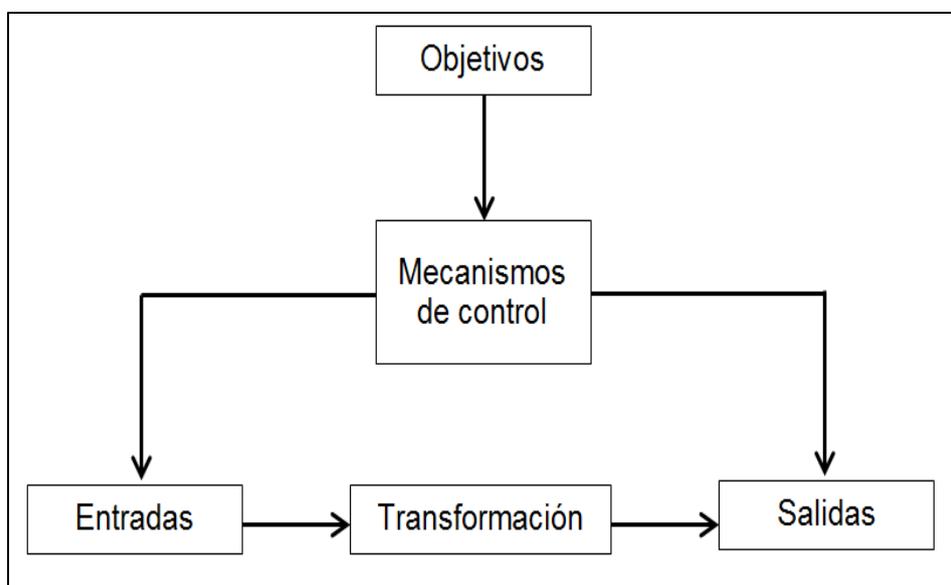
¿Qué es un sistema?

Un sistema es un conjunto de partes interrelacionadas que posee alguna singularidad dentro de su medio. Los sistemas extraen materia prima del medio, la transforman y una vez procesada la introducen de nuevo al ambiente. Todo lo que alimenta a un sistema se denomina “insumo” y todo lo que sale de él se denomina “producto”. En los sistemas de información, los insumos se llaman datos y los productos se llaman información. (BERNAL Niño, 2004, pág. 13)

Un sistema es un conjunto de componentes que interactúan entre sí para lograr un objetivo común. Aunque existen una gran variedad de sistemas, la mayoría de ellos pueden representarse a través de un modelo formado por cinco bloques básicos; elementos de entrada, elementos de salida, sección de transformación, mecanismos de control y objetivos. Tal y como se muestra en la figura 1, los recursos acceden al sistema a través de los elementos de entrada para ser modificados en la sección de transformación. Este proceso es controlado por el mecanismo de control con el fin de lograr el objetivo marcado. Una vez se ha llevado a cabo la

transformación, el resultado sale del sistema a través de los elementos de salida. (FERNÁNDEZ Alarcón, 2006, pág. 11)

Figura N° 1: Modelo General de un Sistema



Definición de sistemas de información

Un sistema de información se puede definir técnicamente como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores a analizar problemas, visualizar asuntos complejos y crear productos nuevos. (LAUDON & LAUDON, 2004, pág. 8)

Los sistemas de información contienen información acerca de gente, lugares y cosas importantes dentro de la organización o en el entorno en que se desenvuelve. Por información se entiende los datos que se han modelado en forma significativa y útil para los

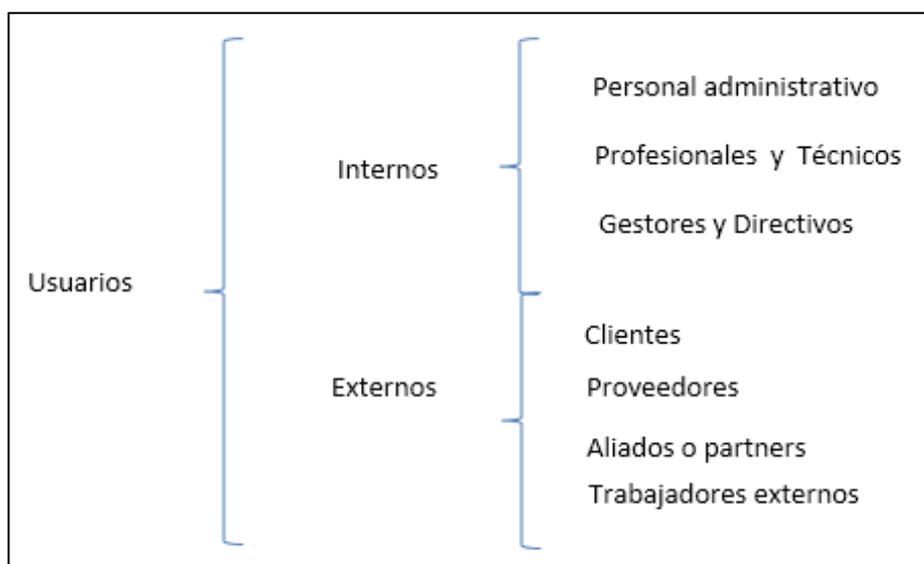
seres humanos. En contraste, los datos son secuencias de hechos en bruto y representan eventos que ocurren en las organizaciones o en el entorno físico antes de ser organizados y ordenados en una forma que las personas puedan entender y utilizar. (LAUDON & LAUDON, 2004, pág. 8)

Usuarios de sistemas

Los usuarios de sistemas son aquellas personas que utilizan los sistemas de información de una forma regular para capturar, introducir, validar, transformar y almacenar datos e información. Entre todos los grupos de individuos que participan en el desarrollo de un sistema de información, los usuarios es el más cuantioso.

Así mismo, los usuarios deben ser considerados como el grupo de individuos más importante en el desarrollo de un sistema de información, ya que será este colectivo el que tendrá que trabajar diariamente sobre él, y el que decidirá si cumple con las necesidades que tiene el negocio.

Los sistemas de información pueden ser utilizados por una gran cantidad de individuos con objetivos y necesidades muy diversas. Es por ese motivo que puede ser interesante agrupar a los usuarios de sistemas en grupos y subgrupos en función de la relación con la empresa. Para empezar, se puede distinguir entre usuarios internos a la organización y usuarios externos a la organización.

Figura N° 2: Tipología de usuarios de sistemas

Diseñadores de sistemas

Los diseñadores de sistemas son expertos en la tecnología que resuelven las necesidades y las restricciones manifestadas por usuarios de la empresa mediante recursos tecnológicos.

Debido al crecimiento en el desarrollo de la tecnología, así como su utilización, los diseñadores de sistemas han sido especializados e lo largo de las últimas décadas. Algunos de estas especialidades son la administración de datos (tecnologías de bases de datos), la arquitectura de redes (tecnologías de comunicación), el diseño Web), o la seguridad (tecnologías de seguridad y privacidad).

A diferencia de los propietarios y de los usuarios de sistemas, los diseñadores se centran en aspectos tecnológicos más que en aspectos de negocio. La divergencia existente entre la perspectiva de los usuarios de sistemas y de los diseñadores de sistemas hace

necesario introducir un nuevo individuo en el desarrollo de sistemas: el analista de sistemas. (WHITTEN, BENTLEY, & DITTMAN, 2004):

Constructores de sistemas

Los constructores de sistemas, otro tipo de especialistas en tecnología, son los encargados de fabricar sistemas de información basados en las especificaciones de diseño obtenidas de los diseñadores de sistemas.

Tal y como ocurre con los diseñadores los avances tecnológicos han llevado a especializar a los constructores en distintas tareas en el desarrollo de sistemas. Entre ellas se pueden nombrar la de programador de aplicaciones informáticas, la de programador de sistemas, la de programador de base de datos o la de integrador de software. (WHITTEN, BENTLEY, & DITTMAN, 2004)

Analista de sistemas

Un analista de sistemas es una persona que estudia los problemas y las necesidades de una empresa para determinar cómo podrían combinarse los recursos humanos, los procesos, los datos y la tecnología de la información para obtener mejoras en la empresa.

Las habilidades necesarias para cumplir de una forma eficiente las funciones de un analista de sistemas son (WHITTEN, BENTLEY, & DITTMAN, 2004):

2.2.2 METODOLOGÍAS DEL DESARROLLO DEL SOFTWARE

Un proceso de software es un conjunto de actividades que conducen a la creación de un producto de software. Estas actividades pueden consistir en el desarrollo de software desde cero en un lenguaje de programación estándar como java o C. Sin embargo, cada vez, mas, se desarrolla nuevo software ampliando y modificando los sistemas existentes y configurando e integrando software comercial o componentes del sistema. (SOMMERVILLE, 2005, pág. 60)

Definición de Metodología

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo.

Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, incremental...). Definen artefactos, roles y actividades, junto con prácticas y técnicas recomendadas.

La metodología para el desarrollo de software en un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que

cumplimos el objetivo por el cual fue creado. (S., PRESSMAN, 2007)

Tipos de Metodologías

Modelo de cascada

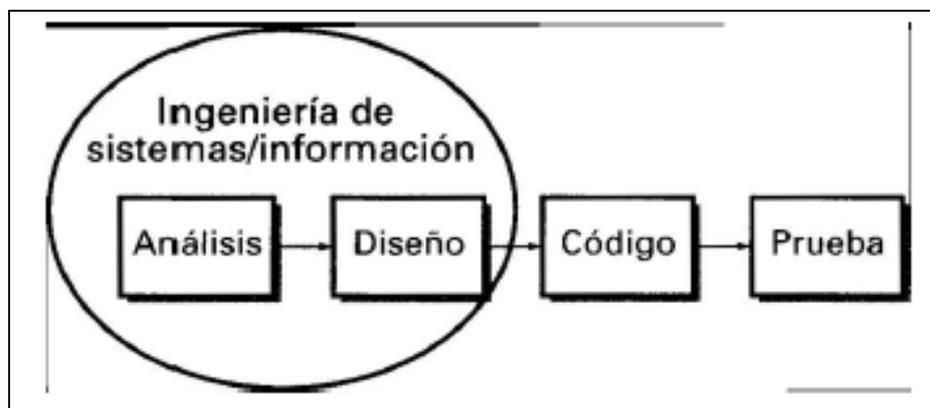
Llamado algunas veces «ciclo de vida básico» o modelo en cascada», el modelo lineal secuencial sugiere un enfoque5 sistemático, secuencial, para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento. La Figura 3 muestra el modelo lineal secuencial para la ingeniería del software. Modelado según el ciclo de ingeniería convencional, el modelo lineal secuencial comprende las siguientes actividades: (S., PRESSMAN, 2007)

Ingeniería y modelado de Sistemas/Información

Como el software siempre forma parte de un sistema más grande (o empresa), el trabajo comienza estableciendo requisitos de todos los elementos del sistema y asignando al software algún subgrupo de estos requisitos. Esta visión del sistema es esencial cuando el software se debe interconectar con otros elementos como hardware, personas y bases de datos. La ingeniería y el análisis de sistemas comprenden los requisitos que se recogen en el nivel del sistema con una pequeña parte de análisis y de diseño. La ingeniería de información abarca los requisitos que se recogen en

el nivel de empresa estratégico y en el nivel del área de negocio.
(S., PRESSMAN, 2007)

Figura N° 3: El modelo lineal secuencial



Análisis de los requisitos del software: El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Para comprender la naturaleza del (los) programa(s) a construirse, el ingeniero («analista») del software debe comprender el dominio de información del software, así como la función requerida, comportamiento, rendimiento e interconexión. (S., PRESSMAN, 2007)

Diseño: El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos de programa: estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental (algoritmo). El proceso del diseño traduce requisitos en una representación del software donde se pueda evaluar su calidad antes de que comience la codificación. (S., PRESSMAN, 2007)

Generación de código: El diseño se debe traducir en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Si se lleva a cabo el diseño de una forma detallada, la generación de código se realiza mecánicamente. (S., PRESSMAN, 2007)

Pruebas: Una vez que se ha generado el código, comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores y asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos. (S., PRESSMAN, 2007)

Mantenimiento: El software indudablemente sufrirá cambios después de ser entregado al cliente (una excepción posible es el software empotrado). Se producirán cambios porque se han encontrado errores, porque el software debe adaptarse para acoplarse a los cambios de su entorno externo (por ejemplo: se requiere un cambio debido a un sistema operativo o dispositivo periférico nuevo), o porque el cliente requiere mejoras funcionales o de rendimiento. El soporte y mantenimiento del software vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo. (S., PRESSMAN, 2007)

Modelo espiral

El modelo de espiral, desarrollado durante la década de los ochenta, es una extensión del modelo de cascada. A diferencia del modelo de cascada, que es dirigido por documentos, el modelo espiral se basa en una estrategia para reducir el riesgo del proyecto en áreas de incertidumbre, como requerimientos iniciales incompletos e inestables. El modelo enfatiza ciclos de trabajo, cada uno de los cuales estudia el riesgo antes de proceder al siguiente ciclo. Cada ciclo comienza con la identificación de los objetivos, soluciones alternativas, restricciones asociadas con cada alternativa y, finalmente, se procede a su evaluación. Cada ciclo del modelo de espiral termina con una revisión que discute los logros actuales y los planes para el siguiente ciclo. (WEITZENFELD, 2005, pág. 53)

El modelo en espiral del proceso del software fue originalmente propuesto por Boehm (Boehm, 1998). Más que representar el proceso del software como una secuencia de actividades con retrospectiva de una actividad a otra, se presenta como una espiral. Así el ciclo más interno podría referirse a la viabilidad del sistema, el siguiente ciclo a la definición de requerimientos, el siguiente ciclo al diseño del sistema se divide en cuatro sectores: (SOMMERVILLE, 2005, págs. 68-69)

1. Definición de objetivos: Para esta fase del proyecto se definen los objetivos específicos. Se identifican las restricciones del

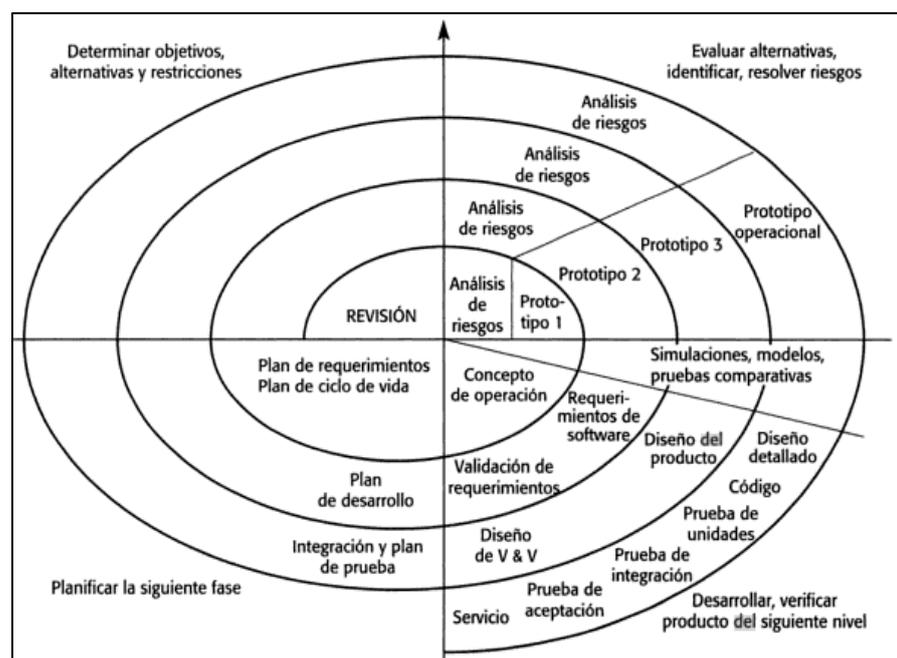
proceso y el producto, y se traza un plan detallado de gestión. Se identifican los riesgos del proyecto. Dependiendo de estos riesgos, se plantean estrategias alternativas.

2. Evaluación y reducción de riesgos: Se lleva a cabo un análisis detallado para cada uno de los riesgos del proyecto identificado. Se definen los pasos para reducir dichos riesgos.

3. Desarrollo y validación: Después de la evaluación de los riesgos, se elige un modelo para el desarrollo del sistema.

4. Planificación: El proyecto se revisa y se toma la decisión de si se debe continuar con un ciclo posterior de la espiral. Si se decide continuar, se desarrollan los planes para la siguiente fase del proyecto. (SOMMERVILLE, 2005, págs. 68-69)

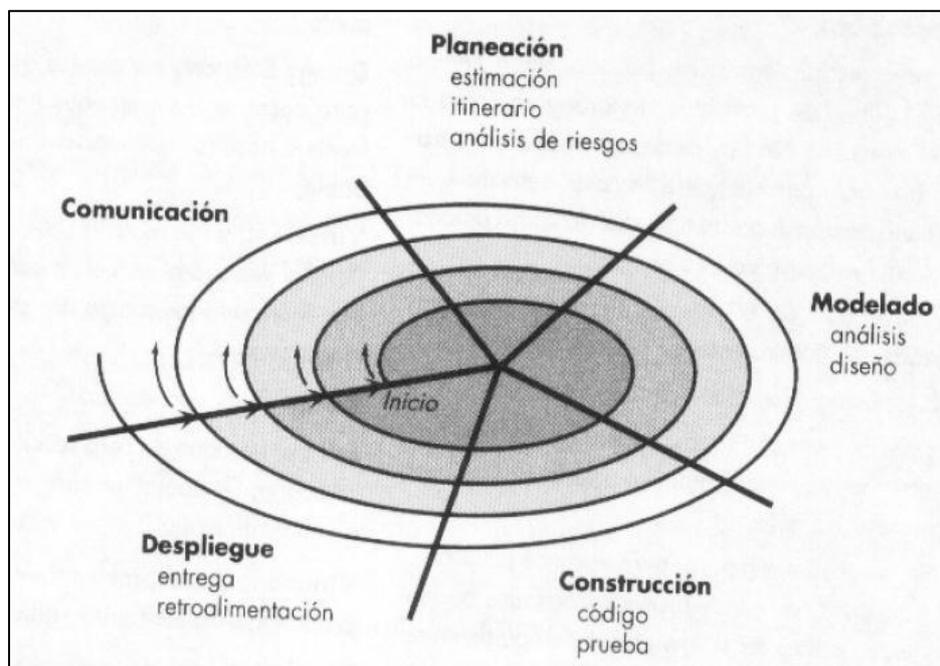
Figura N° 4: Modelo en espiral de Boehm para el proceso del software (IEEE, 1998)



El modelo espiral, que Boehm propuso originalmente, es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de la construcción de prototipos con los aspectos controlados y sistemáticos del modelo en cascada. Proporciona el material para el desarrollo rápido de versiones incrementales del software. Boehm describe este modelo de la siguiente manera:

El modelo de desarrollo en espiral es un generador del modelo de proceso guiado por el riesgo que se emplea para conducir sistemas intensivos de ingeniería de software concurrente y con múltiples usuarios. Tiene dos características distintivas principales. Una de ellas es un enfoque cíclico para el crecimiento incremental del grado de definición e implementación de un sistema, mientras disminuye su grado de riesgo. La otra es un conjunto de puntos de fijación para asegurar el compromiso del usuario con soluciones de sistema que sean factibles y mutuamente satisfactorias. (PRESSMAN, 2006, pág. 58)

Un proceso en espiral se divide en un conjunto de actividades del marco de trabajo que define el equipo de ingeniería del software. Cada una de las actividades genéricas del marco de trabajo representa un segmento de la ruta en espiral que se presenta en la figura 5 Cuando comienza este proceso evolutivo el equipo de software realiza actividades implicadas en un círculo alrededor de la espiral que tiene una dirección en el sentido del movimiento de las manecillas del reloj, y que se inicia desde el centro. (PRESSMAN, 2006, págs. 58-59)

Figura N° 5: Un modelo espiral típico

El modelo en espiral es un enfoque realista para el desarrollo del software y sistemas a gran escala. Como el software evoluciona conforme avanza el proceso, el desarrollador y el cliente entienden y reaccionan de mejor manera ante los riesgos en cada etapa evolutiva. El modelo en espiral emplea la construcción de prototipos en cualquier etapa evolutiva del producto. Mantiene el enfoque sistemático de los pasos que sugiere el ciclo de vida clásico, pero lo incorpora al marco de trabajo iterativo que refleja de forma más verídica el mundo real. El modelo en espiral exige una consideración directa de los riesgos técnicos en todas las etapas del proyecto y, si se aplica en forma apropiada, debe reducir los riesgos antes de que se vuelvan problemáticos. (PRESSMAN, 2006, pág. 59)

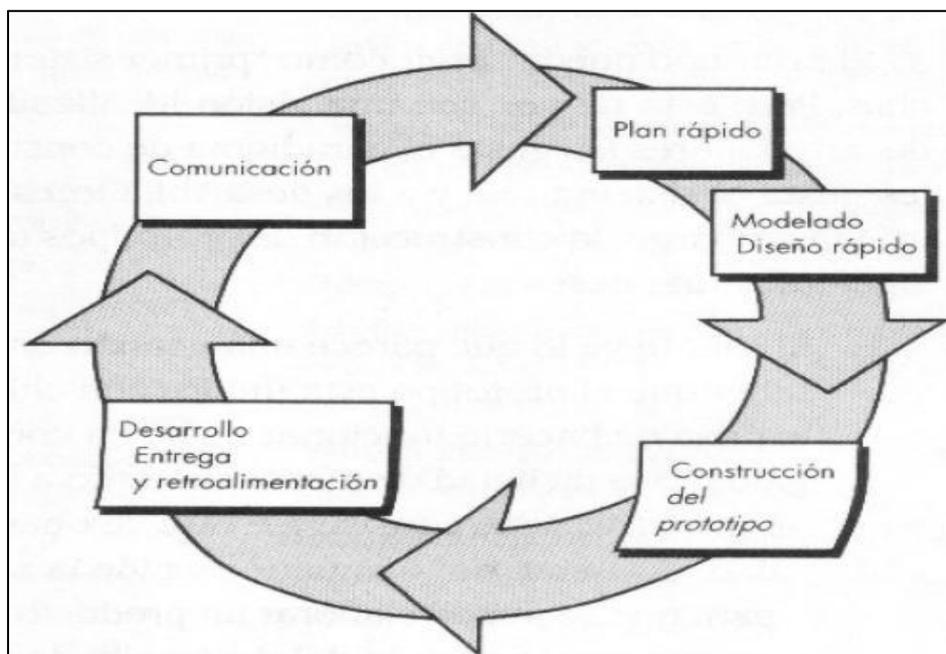
Construcción de Prototipos

Un prototipo es una versión preliminar, intencionalmente incompleta o reducida de un sistema. El uso de prototipo es una estrategia que puede aplicarse en casi todas las actividades del proceso de software. El propósito de los prototipos es obtener rápidamente la información necesaria para ayudar en la toma de decisiones. (WEITZENFELD, 2005, pág. 47)

Kenneth E. Kendall menciona: “La elaboración de prototipos es una técnica de recopilación útil para complementar el ciclo de vida de desarrollo de un sistema tradicional.”

El paradigma de construcción de prototipos (figura 6) se inicia con la comunicación. El ingeniero de software y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de prototipos y se presenta el modelado (en la forma de un diseño rápido). El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. (PRESSMAN, 2006, págs. 55-56)

Figura N° 6: El modelo de construcción de prototipos



El prototipo puede servir como “primer sistema”, el que Brooks recomienda desechar. Pero ésta tal vez sea una visión idealizada. Sin embargo, la construcción de prototipos también se toma problemática por las siguientes razones: (PRESSMAN, 2006, págs. 56-57)

1. El cliente ve lo que parece una versión en funcionamiento del software, sin saber que el prototipo está unido “con chicle y cable para embalaje”, que por la prisa de hacerlo funcionar no se ha considerado la calidad del software global o la facilidad de mantenimiento a largo plazo. Cuando se informa que el producto debe construirse otra vez para mantener los altos niveles de calidad, el cliente no le entiende y pide la aplicación de “unos pequeños ajustes” para que pueda elaborar un producto final a

partir del prototipo. Es muy frecuente que la gestión del desarrollo de software sea muy lenta.

2. A menudo, el desarrollador establece compromisos de implementación para lograr que el prototipo funcione con rapidez. Tal vez se utilice un sistema operativo o lenguaje de programación inadecuado solo porque está disponible y es conocido; se puede implementar un algoritmo ineficiente sólo para mostrar capacidad. Después de un tiempo, el desarrollador quizá se familiarice con estas selecciones y olvide las razones por las que son inapropiadas. La sección menos ideal ahora se convierte en una parte integral del sistema.

A pesar de que tal vez surjan problemas, la construcción de prototipos puede ser un paradigma efectivo para la ingeniería de software. La clave es definir las reglas del juego desde el principio; es decir, el cliente y el desarrollador se deben poner de acuerdo en que el prototipo se construya y sirva como un mecanismo para la definición de requisitos, es que se descarte, al menos en parte, y que después se desarrolle el software real con un enfoque hacia la calidad. (PRESSMAN, 2006, pág. 57)

Lenguaje Unificado de Modelado

El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional, Rational Unified Process o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que programación estructurada es una forma de programar como lo es la orientación a objetos, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML solo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

Tipos de diagramas en UML

Estructurales.- Muestran la estructura estática de los objetos en un sistema.

Diagrama de clases: Los diagramas de clase son, sin duda, el tipo de diagrama UML más utilizado. Es el bloque de construcción principal de cualquier solución orientada a objetos. Muestra las clases en un sistema, atributos y operaciones de cada clase y la relación entre cada clase. En la mayoría de las herramientas de modelado, una clase tiene tres partes, nombre en la parte superior, atributos en el centro y operaciones o métodos en la parte inferior. En sistemas grandes con muchas clases relacionadas, las clases se agrupan para crear diagramas de clases. Las Diferentes relaciones entre las clases se muestran por diferentes tipos de flechas.

Diagrama de componentes: Un diagrama de componentes muestra la relación estructural de los componentes de un sistema de software. Estos se utilizan principalmente cuando se trabaja con sistemas complejos que tienen muchos componentes. Los componentes se comunican entre sí mediante interfaces. Las interfaces se enlazan mediante conectores.

Diagrama de despliegue: Un diagrama de despliegue muestra el hardware de su sistema y el software de ese hardware. Los diagramas de implementación son útiles cuando la solución de software se despliega en varios equipos, cada uno con una configuración única.

Diagrama de objetos: Los diagramas de objetos, a veces denominados diagramas de instancia, son muy similares a los diagramas de clases. Al igual que los diagramas de clases, también muestran la relación entre los objetos, pero usan ejemplos del mundo real. Se utilizan para mostrar cómo se verá un sistema en un momento dado. Debido a que hay datos disponibles en los objetos, a menudo se utilizan para explicar relaciones complejas entre objetos.

Diagrama de paquetes: Como su nombre indica, un diagrama de paquetes muestra las dependencias entre diferentes paquetes de un sistema.

Diagrama de perfiles El diagrama de perfil es un nuevo tipo de diagrama introducido en UML 2. Este es un tipo de diagrama que se utiliza muy raramente en cualquier especificación.

Diagrama de estructura compuesta: Los diagramas de estructura compuesta se utilizan para mostrar la estructura interna de una clase.

De comportamiento: Muestran el comportamiento dinámico de los objetos en el sistema.

Diagrama de actividades: Los diagramas de actividad representan los flujos de trabajo de forma gráfica. Pueden utilizarse para describir el flujo de trabajo empresarial o el flujo de trabajo operativo de cualquier componente de un sistema. A veces, los diagramas de actividad se utilizan como una alternativa a los diagramas de máquina del estado.

Diagrama de casos de uso: Como el tipo de diagrama de diagramas UML más conocido, los diagramas de casos de uso ofrecen una visión general de los actores involucrados en un sistema, las diferentes funciones que necesitan esos actores y cómo interactúan estas diferentes funciones. Es un gran punto de partida para cualquier discusión del proyecto, ya que se pueden identificar fácilmente los principales actores involucrados y los principales procesos del sistema.

Diagrama de máquina de estados: Los diagramas de máquina de estado son similares a los diagramas de actividad, aunque las anotaciones y el uso cambian un poco. En algún momento se conocen como diagramas de estados o diagramas de diagramas de estado también. Estos son muy útiles para describir el comportamiento de los objetos que actúan de manera diferente de acuerdo con el estado en que se encuentran en el momento.

De interacción

Diagrama global de interacciones: Los diagramas generales o globales de interacción son muy similares a los diagramas de actividad. Mientras que los diagramas de actividad muestran una secuencia de procesos, los diagramas de interacción muestran una secuencia de diagramas de interacción. En términos simples, pueden llamarse una colección de diagramas de interacción y el orden en que suceden. Como se mencionó anteriormente, hay siete tipos de diagramas de interacción, por lo que cualquiera de ellos puede ser un nodo en un diagrama de vista general de interacción.

Diagrama de comunicación: El diagrama de comunicación se llamó diagrama de colaboración en UML. Es similar a los diagramas de secuencia, pero el foco está en los mensajes pasados entre objetos.

Diagrama de secuencia: Los diagramas de secuencia en UML muestran cómo los objetos interactúan entre sí y el orden en que se producen esas interacciones. Es importante tener en cuenta que muestran las interacciones para un escenario en particular. Los procesos se representan verticalmente y las interacciones se muestran como flechas.

Diagrama de tiempos: Los diagramas de sincronización son muy similares a los diagramas de secuencia. Representan el comportamiento de los objetos en un marco de tiempo dado. Si es solo un objeto, el diagrama es directo, pero si hay más de un objeto

involucrado, también se pueden usar para mostrar interacciones de objetos durante ese período de tiempo.

Los diagramas de secuencia de UML forman parte de un modelo UML y solo existen dentro de los proyectos de modelado UML. Para crear un diagrama de secuencia UML, en el menú Arquitectura, haga clic en Nuevo diagrama de capas o UML. Obtenga más información sobre elementos de diagrama de secuencia UML o diagramas de modelado UML en general. Para ver una demostración en vídeo, consulte Esbozar interacciones mediante diagramas de secuencia (2010).

2.2.3 SISTEMAS GESTORES DE BASES DE DATOS

Concepto de base de datos

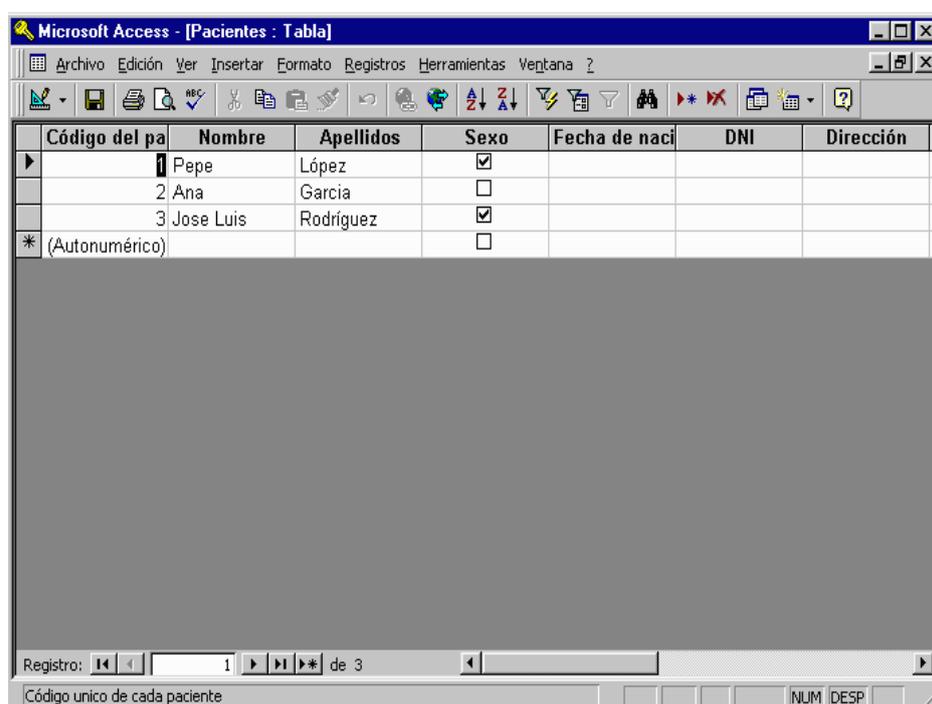
Una base de datos es un programa computacional que permite almacenar y gestionar gran cantidad de información. En una base de datos la información se organiza en registros y campos.

Los campos pueden ser de varios tipos. Campo de tipo carácter, campo de tipo numérico y campo de tipo lógico. (BALLÚS, FORNALS, & GALVE, 2004, pág. 320)

La base de datos se utiliza para referirse a una gran masa de datos que se hallan relacionados entre sí. Estos datos se encuentran divididos en varias categorías, que son los registros, los ficheros, las bibliotecas, etc.

La base de datos no es más que un potente manipulador de las relaciones existentes entre estas jerarquías de información, con el que se puede definir nuevas relaciones o acceder a los datos mediante las ya definidas. (BIOSCA, GÁRRIZ, & SORT, 2005, pág. 1086)

Figura N° 7: Pantalla de una base de datos



Código del pa	Nombre	Apellidos	Sexo	Fecha de naci	DNI	Dirección
1	Pepe	López	<input checked="" type="checkbox"/>			
2	Ana	García	<input type="checkbox"/>			
3	Jose Luis	Rodríguez	<input checked="" type="checkbox"/>			
*	(Autonumérico)		<input type="checkbox"/>			

Tipos de base de datos

Modelo Entidad-Relación

Está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre esos objetos. Una entidad es una <cosa>, u <objeto> en el mundo real que es distinguible de otros objetos. Por ejemplo, una

persona es una entidad, y las cuentas bancarias pueden ser consideradas entidades.

Las entidades se describen en una base de datos mediante un conjunto de atributos. Una relación es una asociación entre varias entidades. El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente conjunto de entidades y conjunto de relaciones. (SIBERSCHATZ & F. Korth , 2002, pág. 20)

El modelo Entidad-Relación consta de los siguientes componentes:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombo, que representan relaciones entre conjuntos de entidades.
- Líneas, unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

Modelo Relacional

Se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único.

Otros modelos de Datos

El modelo orientado a objetos se puede observar como una extensión del modelo E-R con las nociones de encapsulamiento, métodos (funciones) e identidad de objeto.

El modelo de datos relacional orientado a objetos combina las características del modelo de datos orientado a objetos y el modelo de datos relacional. (SIBERSCHATZ & F. Korth , 2002, pág. 21 y 22)

Componentes de los sistemas gestores de bases de datos

Los SGBD son paquetes de software muy complejos que deben proporcionar una serie de servicios que van a permitir almacenar y explotar los datos de forma eficiente. Los componentes principales son los siguientes:

Lenguajes de los SGBD. Los lenguajes van a permitir al administrador de la BD especificar los datos que componen la BD, su estructura, las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios. Los lenguajes de los SGBD se clasifican en:

- Lenguaje de definición de datos (DDL o LDD)
- Lenguaje de manipulación de datos (LMD o DML)

Diccionario de datos. Es el lugar donde se deposita información acerca de todos los datos que forman la BD. Es una guía en la que se describe la BD y los objetos que la forman.

Seguridad e integridad de datos. Un SGBD proporciona los siguientes mecanismos para garantizar la seguridad e integridad de los datos:

- Garantizar la protección de los datos contra accesos no autorizados.
- Ofrecen mecanismos para implantar restricciones de integridad de la BD.
- Proporciona herramientas para la planificación y realización de copias de seguridad y restauración.

Administrador de la BD. Existen diferentes categorías de usuarios:

- Usuarios de la categoría DBA. Administran la base de datos, tienen el nivel más alto de privilegios.
- Usuarios categoría RESOURCE, pueden crear sus propios objetos.
- Usuarios tipo CONNECT, solo utilizan objetos a los que tienen permisos. (RAMOS, RAMOS, & MONTERO, 21-10-2008, pág. 11)

Arquitectura de los sistemas de Bases de datos

Un sistema de bases de datos se divide en módulos que se encargan de cada una de las responsabilidades del sistema

completo. Los componentes funcionales de un sistema de bases de datos se pueden dividir a grandes rasgos en los componentes gestores de almacenamientos y procesador de consultas. (BIOSCA, GÁRRIZ, & SORT, 2005, pág. 1087)

Gestor de Almacenamiento. Es un módulo de programa que proporciona la interfaz entre los datos de bajo nivel en la base de datos y los programas de aplicación y consultas emitidas al sistema. Es responsable de la interacción con el gestor de archivos. Traduce las diferentes instrucciones LMD a órdenes de un sistema de archivos de bajo nivel. Es responsable del almacenamiento, recuperación y actualización de los datos en la base de datos.

Los componentes del gestor de almacenamiento incluyen (BALLÚS, FORNALS, & GALVE, 2004, pág. 321):

- Gestor de autorización e integridad
- Gestor de transacciones
- Gestor de archivos
- Gestor de memoria intermedia

El gestor de almacenamiento implementa varias estructuras de datos como parte de la implementación física del sistema.

- Archivos de datos
- Diccionario de datos
- Índices

Procesador de consultas. Los componentes del procesador de consultas incluyen (SIBERSCHATZ & F. Korth , 2002, pág. 24):

- Interprete del LDD
- Compilador del LMD
- Motor de evaluación de consultas.

2.2.4 INTRODUCCIÓN AL MICROSOFT SQL SERVER

¿Qué es Microsoft SQL Server?

Es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es Transact-SQL(TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

Características

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DMLgráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.

- Además permite administrar información de otros servidores de datos.

Es común desarrollar proyectos completos empleando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa la base de datos (Microsoft SQL Server), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

En el manejo de SQL mediante líneas de comando se utiliza el SQLCMD, osql, o PowerShell.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos.

El tipo NUMERIC fue mejorado para ser usado como identificador de columna a partir de la versión 2008 R2.

Interfaz de Usuario

SQL Server proporciona unos interfaz que han cambiado durante los años, de los cuales los más conocidos son los interfaz gráficos que están utilizados como herramienta de desarrollo estándar a los desarrolladores y administradores.

La interfaz gráfica hasta 2005 incluyó el Enterprise Manager con una vista de árbol de los distintos objetos y con la capacidad de manejarlos; y el Query analyzer como interfaz textual para ejecutar comandos de TSQL.

Aparte de los interfaces estándares de SQL Server, se puede ejecutar comandos de TSQL con herramientas de conexión como ODBC y OLE-DB.¹²

2.2.5 VISUAL BASIC

Visual Basic (VB) es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo que facilitó en cierta medida la programación misma.

La última versión fue la 6, liberada en 1998, para la que Microsoft extendió el soporte hasta marzo de 2008.

En 2001 Microsoft propuso abandonar el desarrollo basado en la API Win32 y pasar a un framework o marco común de librerías, independiente de la versión del sistema operativo .NET Framework, a través de Visual Basic .NET (y otros lenguajes como C Sharp(C#) de fácil transición de código entre ellos); fue el sucesor de Visual Basic 6.

Aunque Visual Basic es de propósito general, también provee facilidades para el desarrollo de aplicaciones de bases de datos usando Data Access Objects, Remote Data Objects o ActiveX Data Objects.

Visual Basic contiene un entorno de desarrollo integrado o IDE que integra editor de textos para edición del código fuente, un depurador, un compilador (y enlazador) y un editor de interfaces gráficas o GUI.

Visual Basic dio un paso más en innovación y ahora es posible programar aplicaciones Nativas para Android o iPhone utilizando un software de la compañía Anywhere Software que transforma código Visual Basic (creado por dicha compañía) en APPs 100 % nativas en Java para ambos sistemas operativos de dispositivos móviles.

Características

Los compiladores de Visual Basic generan código que requiere una o más librerías de enlace dinámico para que funcione, conocidas comúnmente como DLL (sigla en inglés de Dynamic-Link Library); en algunos casos reside en el archivo llamado MSVBVMxy.DLL (siglas de "MicroSoft Visual Basic Virtual Machine x.y", donde x.y es la versión) y en otros en VBRUNXXX.DLL ("Visual Basic Runtime X.XX"). Estas bibliotecas DLL proveen las funciones básicas implementadas en el lenguaje, conteniendo rutinas en código ejecutable que son cargadas bajo demanda en tiempo de

ejecución. Además de las esenciales, existe un gran número de bibliotecas del tipo DLL con variedad de funciones, tales como las que facilitan el acceso a la mayoría de las funciones del sistema operativo o las que proveen medios para la integración con otras aplicaciones.

El propio Visual Basic provee soporte para empaquetado y distribución; es decir, permite generar un módulo instalador que contiene al programa ejecutable y las bibliotecas DLL necesarias para su ejecución. Con ese módulo la aplicación desarrollada se distribuye y puede ser instalada en cualquier equipo (que tenga un sistema operativo compatible).

Así como bibliotecas DLL, hay numerosas aplicaciones desarrolladas por terceros que permiten disponer de variadas y múltiples funciones, incluso mejoras para el propio Visual Basic; las hay también para el empaquetado y distribución, y hasta para otorgar mayor funcionalidad al entorno de programación (IDE).

Objetos y eventos

Se designa como objeto cualquier elemento, por ejemplo, un formulario, una imagen, un control, tal como una caja de texto; a su vez, los objetos tienen propiedades, que en el caso de la caja de texto una es la propiedad "text" que se encarga de contener el texto que aparecerá en la caja. A los objetos se les puede asociar eventos. Un evento es la ocurrencia de un suceso, comúnmente la acción que realiza el usuario sobre el objeto, que como resultado

puede, por ejemplo, provocar un cambio en alguna propiedad de un objeto. Por ejemplo: Visual Basic tiene un evento llamado KeyPress, que ocurre cuando el usuario presiona una tecla; ese evento se puede asociar a la caja de texto, y en él definirá (por programación) qué acción se tomará cuando se oprima una tecla.

En síntesis, un objeto posee propiedades, responde a eventos y puede ejecutar métodos asociados a él.

2.2.6 INTERFAZ DE USUARIO

La interfaz de usuario es el espacio donde se producen las interacciones entre seres humanos y máquinas. El objetivo de esta interacción es permitir el funcionamiento y control más efectivo de la máquina desde la interacción con el humano.

Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, contenido gráfico, cursor, los beeps y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora.

El objetivo del diseño de una interfaz es producir una interfaz que sea fácil de usar (explicarse por sí misma), eficiente y agradable para que al operar la máquina dé el resultado deseado.

Tipos.- En las interfaces de usuario se pueden distinguir básicamente tres tipos:

1. **Una interfaz de hardware:** a nivel de los dispositivos utilizados para ingresar, procesar y entregar los datos: teclado, ratón y pantalla visualizadora.
2. **Una interfaz de software:** destinada a entregar información acerca de los procesos y herramientas de control, a través de lo que el usuario observa habitualmente en la pantalla.
3. **Una interfaz de software-hardware:** que establece un puente entre la máquina y las personas, permite a la máquina entender la instrucción y al hombre entender el código binario traducido a información legible.

Según su construcción

Pueden ser de hardware o de software:

- **Interfaces de hardware:** Se trata de un conjunto de controles o dispositivos que permiten que el usuario intercambie datos con la máquina, ya sea introduciéndolos (pulsadores, botones, teclas, reguladores, palancas, manivelas, perillas) o leyéndolos (pantallas, diales, medidores, marcadores, instrumentos).
- **Interfaces de software:** Son programas o parte de ellos, que permiten expresar las órdenes a la computadora o visualizar su respuesta.

2.2.7 ISO 9126

ISO 9126 es un estándar internacional para la evaluación de la calidad del software. Está reemplazado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos.

El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso y expendido.

El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

Funcionalidad.- Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.

- Idoneidad.
- Exactitud.
- Interoperabilidad.
- Seguridad.
- Cumplimiento de normas.

Fiabilidad.- Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.

- Madurez.
- Recuperabilidad.
- Tolerancia a fallos

Usabilidad.- Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

- Aprendizaje.
- Comprensión.
- Operatividad.
- Atractividad.

Eficiencia.- Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.

- Comportamiento en el tiempo.
- Comportamiento de recursos.

Mantenibilidad.- Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.

- Estabilidad.

- Facilidad de análisis.
- Facilidad de cambio.
- Facilidad de pruebas.

Portabilidad.- Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra.

- Capacidad de instalación.
- Capacidad de reemplazamiento.
- Adaptabilidad.
- Co-Existencia.

La subcaracterística Conformidad no está listada arriba ya que se aplica a todas las características. Ejemplos son conformidad a la legislación referente a usabilidad y fiabilidad.

Cada subcaracterística (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así. Como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas).

Métricas externas son aquellas aplicables al software en ejecución.

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales.

Idealmente, la calidad interna no necesariamente implica calidad externa y esta a su vez la calidad en el uso.

Este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

- Factores (especificar): Describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Se definen y se usan para proveer una escala y método para la medida.

ISO 9126 distingue entre fallo y no conformidad. Un fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es el incumplimiento de los requisitos especificados. Una distinción similar es la que se establece entre validación y verificación.

2.2.8 PRUEBAS O MÉTRICAS DEL SOFTWARE

La prueba de software es un elemento crítico para la garantía de calidad del software y representa una visión final de las especificaciones del diseño y de la codificación.

La creciente inclusión del software como un elemento más de muchos sistemas y la importancia de los costos adicionales a un fallo del mismo, están motivando la creación de pruebas minuciosas y bien planificadas. La métrica intenta obtener un conjunto de medidas indirectas que dan lugar a métricas que proporcionan una indicación de la calidad de algún tipo de representación del software.

El objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolos con la menor cantidad de tiempo y de esfuerzo.

Las normas que pueden servir para objetos de prueba son:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.

- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si se descubre un error no detectado hasta entonces.

Si la prueba se lleva a cabo con éxito (de acuerdo con el objetivo anteriormente establecido) descubrirá errores en el software. Como ventaja secundaria, la prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento (Roger Pressman, Pág. 302).

Pruebas no convencionales

Son pruebas que no pueden ejecutarse o probarse en el sentido convencional, sin embargo pueden usarse las revisiones técnicas formales para examinar la corrección y consistencia en la fase de análisis y diseño del sistema.

Pruebas convencionales

Son pruebas que pueden ejecutarse o probarse en un sistema y se realizan en la fase de implementación del sistema los cuales son:

- **Caja Blanca:** La prueba de caja blanca denominada a veces prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño para obtener

los casos de prueba. Mediante el método de prueba de caja blanca, se pueden obtener casos de prueba que:

1. Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
 2. Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
 3. Ejecuten todos los bucles en sus límites y con sus límites operacionales.
 4. Ejerciten las estructuras internas de datos para asegurar su validez.
- **Caja Negra:** La prueba de la caja negra se centra en los requisitos funcionales del software; es decir esta prueba permite obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa.

La prueba de caja negra no es una alternativa a la técnica de prueba de la caja blanca, sino por el contrario, se trata de un enfoque complemento que intenta descubrir diferentes tipos de errores.

La prueba de caja negra intenta encontrar errores como:

- Funciones incorrectas o ausentes.
- Errores de interfaz.

Errores de estructura de datos o en acceso a bases de datos externos.

- Errores de rendimiento.
- Errores de inicialización y determinación.

Métricas del Sistema

La métrica intenta obtener un conjunto de medidas indirectas que dan lugar a métricas que proporciona una indicación de la calidad de algún tipo de representación del software.

Métricas de Complejidad

Son todas las métricas de software que definen de una u otra forma la medición de la complejidad; tales como el volumen, tamaño, anidaciones, costo/estimación, agregación y flujo. Estos son los puntos críticos de la concepción, viabilidad, análisis y diseño del software.

Métricas Basadas en la Función

Llamadas también puntos de función (PF), utilizan una medida de funcionalidad íntegra por la aplicación como valor de normalización. Ya que la funcionalidad no se puede medir directamente, se debe derivar indirectamente mediante otras medidas directas. Los puntos de función se derivan con una relación empírica según las medidas contables/directas del dominio de información del software y las evaluaciones de la complejidad del software.

Las características de dominios de información son:

- **Número de entradas del operador:** Se cuenta cada entrada del operador que proporciona diferentes datos orientados a la aplicación.
- **Número de salidas del operador:** Se cuenta cada salida al operador información orientada a la aplicación.
- **Número de peticiones del operador:** Una de las peticiones/consulta se define con una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida.
- **Número de archivos:** Se encuentra cada archivo maestro lógico que interactúa con la aplicación.
- **Número de interfaces externas:** Se cuenta con toda la interfaz legible por la máquina que se utiliza para transmitir información a otro sistema.

Métricas de Calidad

El concepto de métrica es el término que describe muchos y muy variados casos variados casos de medición. Siendo una métrica una medida estadística (no cuantitativa como en otras disciplinas ejemplo la física) que se aplica a todos los aspectos de calidad de software, los cuales deben ser medidos desde diferentes puntos de vista como el análisis, construcción, funcional, documentación, métodos, procesos, usuarios, entre otros.

Integridad

Mide la habilidad de un sistema para resistir ataques (tanto accidentales como intencionados).

- ✓ El ataque puede producirse en cualquier componente del software.
- ✓ Para medir la integridad se debe medir la seguridad y la amenaza, las cuales se estiman o deducen de la evidencia empírica.

Amenaza

Probabilidad de que se produzca un ataque de tipo determinado en un momento determinado.

Seguridad

Probabilidad de que se pueda repeler el ataque de un tipo determinado en un monto determinado.

MÉTRICA PUNTO FUNCIÓN

Miden a una aplicación desde una perspectiva del usuario, dejando de lado los detalles de codificación. Evalúa con absoluta fiabilidad los siguientes aspectos:

1. El valor comercial de un sistema para el usuario
2. Tamaño del proyecto, coste y tiempo de desarrollo
3. Calidad y productividad del programador MIS
4. Esfuerzo de adaptación, modificación y mantenimiento
5. Posibilidad de desarrollo propio
6. Beneficios de implementación en 4GL.

La fórmula de cálculo es la siguiente:

$$PF = CUENTA_TOTAL * (0.65 + 0.01 * \sum f_i)$$

Dónde:

PF : Punto de función.

CUENTA_TOTAL : Total de Puntos de función sin ajustar

(Complejidad). f_i : Total suma de factores de complejidad.

Complejidad

Se identifican las funciones disponibles para el usuario y se organizan en cinco grupos.

Parámetros de Medición	Simple		Media		Compleja		Total
	Cantidad	* Peso	Cantidad	* Peso	Cantidad	* Peso	
Entradas		* 3		* 4		* 6	
Salidas		* 4		* 5		* 7	
Consultas		* 3		* 4		* 6	
F. Lógicos o Internos		* 7		* 10		* 15	
F. de Interfaz		* 5		* 7		* 10	
Total puntos de función sin ajustar (PFSA)							

Factores de complejidad.- Son catorce los factores de complementan la visión externa de la aplicación; pero no están recogidos en la funcionalidad de la aplicación (toman un valor entre 0 - 5).

- 0 = Sin influencia, factor no presente
- 1 = Influencia insignificante, muy baja
- 2 = Influencia moderada o baja
- 3 = Influencia media, normal

- 4 = Influencia alta, significativa
- 5 = Influencia muy alta, esencial

Factores de complejidad a evaluarse en un software

Nº	FACTOR DE COMPLEJIDAD	VALOR (0..5)
1	Comunicación de Datos.	
2	Proceso Distribuido.	
3	Rendimiento	
4	Configuración Operacional compartida	
5	Ratio de Transacciones	
6	Entrada de Datos EN-LÍNEA	
7	Eficiencia con el Usuario Final	
8	Actualizaciones EN-LÍNEA	
9	Complejidad del Proceso Interno	
10	Reusabilidad del Código	
11	Contempla la Conversión e Instalación	
12	Facilidad de Operación (back up, etc.)	
13	Instalaciones Múltiples	
14	Facilidad de Cambios	
FACTOR DE COMPLEJIDAD TOTAL (FCT)		VALOR

2.3 DEFINICIÓN DE TÉRMINOS BÁSICOS

PERSONAL: El término personal refiere varios usos simultáneos. En términos generales, personal refiere a aquello perteneciente a la persona o aquello que es propio de ella. Por ejemplo, las cualidades propias de una persona serán lo que la haga especial, única e irrepetible en el mundo.

TAREAJE: El Tareaje nos muestra todos los datos como el mantenimiento, procesos, consultas y reportes que se van ingresando incluyendo las marcas que se ingresan automáticamente por medio de la interface sea esta (Tarjeta Magnética, Tarjeta por Código de Barras, Biométrico, Hand Punch etc.) Estas marcas se muestran en el tareaje de una manera consolidada y resumida para el mejor entendimiento de la misma. También muestra una Barra de Colores que nos indican todas las opciones que se muestran en el Tareaje y que visualmente nos ayudan a entender mejor los diferentes estados que cada trabajador. Entre las opciones que el sistema nos permite desarrollar el ingreso de marcas Manuales en caso que no se cuente con una interface o si se va a ingresar dicha marca por algún caso extremo. El Ingreso de Marcas nos muestra la pantalla para que nosotros podamos ver la marca determinada del trabajador seleccionado en el día seleccionado, en caso tenga marcas por otros medios aparecerán en esta pantalla con una pequeña descripción en el campo Estado.

Control de Asistencias: Proceso e instrumentos del registro y control de colaboradores

Es el procedimiento administrativo, que consiste en la puesta en práctica de una serie de Instrumentos, con la finalidad de registrar y controlar al personal que labora en una determinada empresa o institución.

Para que las acciones o actividades empresariales se cumplan, es necesario que haya un adecuado registro y control del capital intelectual. Con el control y registro del personal, se trata de asegurar que las diversas unidades de la organización marchen de acuerdo con lo previsto. Los

objetivos centrales de esta técnica es controlar las entradas y salida del personal, cumplimiento del horario de trabajo, controlar horas extras, permisos, vacaciones tardanzas, licencias, etc.

Este proceso técnico se aplica desde el momento en que el colaborador ingresa a laborar a la institución, ya que su ingreso debe registrarse en una ficha personal pre elaborada por el área de desarrollo de recursos humanos, complementándose inmediatamente con su tarjeta de asistencia diaria.

Los instrumentos técnicos de registro y control del desarrollo del recurso humano, serán establecidos de acuerdo a las necesidades, naturaleza y exigencias de la empresa o institución, siendo dentro de los más utilizados los siguientes:

- La Ficha Personal.
- Control Diario de Asistencia.
- Parte Diario de Asistencia.
- Papeletas de Autorización de Salidas.
- Control de Récord laboral.
- File Personal
- Rol Vacacional
- Rol de Cambio de Vigilancia
- Cuadro de Asignación de Personal
- Reglamento Interno de Trabajo

Registro de Colaboradores: El registro de personal es una función técnica importante para llevar a cabo la política de los recursos humanos, se lleva

específicamente en el file o carpeta personal y, ficha familiar; Este proceso técnico comienza en forma sucesiva, apenas el colaborador ingresa a laborar, ya que su ingreso debe registrarse en una ficha o tarjeta pre – elaborada por el área de desarrollo de recursos humanos.

El registro y control de personal constituye una fuente importante de consulta sobre datos personales del colaborador, asistencia e inasistencia, puntualidad, vacaciones, licencias, permisos, ascensos y promociones entre otros, por tal motivo tiene que ser actualizada permanentemente.

Ausentismo: Las faltas al trabajo no programadas originan problemas administrativos serios. La planificación se reduce a nada por la ausencia de un número significativo de la fuerza laboral.

Los esquemas de pago por enfermedad, en ocasiones se contrastan debido a que influyen en quienes no están realmente enfermos para que permanezcan en casa, pues se elimina el miedo de perder las ganancias. Sin embargo, el asunto es más complicado de lo que parece. Es posible que haya más enfermedades entre ciertos grupos de trabajadores debido a la naturaleza de su actividad. El personal que se paga por hora puede tener necesidad de tomar días por enfermedades falsas, ya que a diferencia del personal que se liquida en forma mensual, ellos son supervisados más rigurosamente y sólo pueden atender sus problemas personales de esta manera. Aunque es fácil acusar a alguien de fingirse enfermo si toma algunos días para faltar, por razones poco convincentes, la persona afectada puede estar bajo algunas forma de estrés o simplemente tener un sentimiento de intranquilidad general, cansancio o fatiga. El ausentismo

también puede ser una válvula de seguridad que previene un serio desasosiego industrial, ya que permite a los individuos una manera de expresar una señal de protesta.

Existen algunas tendencias identificadas con más claridad respecto al ausentismo. El alcoholismo es una causa importante de ausencias los lunes. Existen razones ocupacionales para las enfermedades, con frecuencias los conductores sufren de úlceras y enfermedades del aparato digestivo, asimismo, las “enfermedades del avión” han llegado a ser aceptadas como una razón para las inasistencias de los ejecutivos que realizan viajes constantemente por el mundo. El estrés de cada trabajo no es obvio para los observadores externos.

Si los gerentes están irritados por las acumulaciones de enfermedad, deben distinguir a los que realmente están enfermos de los que no lo están. La enfermedad no tiene definición precisa, por ello, es mejor dejar que los gerentes y supervisores traten los casos individuales. Quizá el método más útil sea tratar de crear las condiciones bajo las cuales los empleados deseen ir a trabajar, y esperar la experiencia en lugar de temerla (SHAUN, Tyson, 2004, p279.).

El Control como Fase del Proceso Administrativo: El proceso del control consiste en varios pasos definidos que son fundamentales para todo control administrativo. Lo anterior significa que se aplica el mismo proceso básico de control, independientemente de la actividad de que se trate (Stoner, 1996).

El control se utiliza en cualquier organización, sea grande o pequeña, y su objetivo es doble: se trata, en primer lugar, de conocer o si se están realizando las actividades planificadas o si se están cumpliendo las órdenes impartidas. Si no ocurre así, el control, en segundo lugar, tiene el objetivo de introducir los correctivos y los ajustes adecuados. A través del control, los directivos se mantienen informados de cómo marchan las actividades en los servicios, se vigila el cumplimiento de las labores por parte del personal.

El propósito último y fundamental del control es el de identificar las fallas y los errores, corregirlos y evitar que vuelvan a repetirse. Para que el control sea efectivo, debe ser oportuno y seguido de correcciones, que pueden adoptar muchas formas distintas incluyendo las sanciones al personal que incumple irresponsablemente sus actividades, tareas y obligaciones (Koontz, 1994).

Control: Para Fayol, (1981), el control es vigilar para que todas las acciones y operaciones se realicen de acuerdo a los planes adoptados, a los principios establecidos y a las normas implantadas. Se aplica a todo, a las personas, a las cosas y a los actos que se cumplen dentro del organismo y está presente en todas las funciones de la empresa. El control debe establecerse como sistema.

Para Terry (1986), el control es el proceso para determinar lo que se está realizando, valorizando y aplicando las medidas de corrección para que lo ejecutado esté de acuerdo con lo planificado. El autor explica que el control no se establece, en forma explícita, como sistema.

Para Koontz y O Donnell (1994), el control “es la función de medición y corrección de las ejecutorias de los subordinados de tal manera que se asegure el cumplimiento de los objetivos y planes del organismo o empresa”.

Tipos de control: Melinkoff (1987) establece que existen clasificaciones del control que se hacen atendiendo a diversos criterios, entre las que se cuentan:

a) De acuerdo a la oportunidad con que se practica, en control puede ser.

- **Previo o a priori:** en este caso, el control se realiza antes de que la acción se cumpla; algunos los critican porque dicen que le resta dinamismo a la acción, le quita la iniciativa a las personas. Es un control que se practica para evitar la contravención a las normas y para canalizar las actividades de la forma que se considere más conveniente.

- **Posterior:** en este otro caso, el control se realiza después que la acción se ha cumplido. Se basa en un conjunto de normas que el personal debe conocer, aceptar y respetar. Por lo tanto, el personal debe estar dispuesto a cumplirlas. Si el personal sabe que lo van a controlar, hace mejor sus labores, se esmera en ser más dinámico y más ágil.

b) De acuerdo a la cobertura de la materia que se controla, el control puede ser:

- **Total:** en este caso, el control se aplica a toda la materia, a todos actos, hechos y personas.

- **Selectivos:** en esta cobertura, el control se aplica sólo a una parte de la materia, actos, hechos y personas; en esta forma es de bajo costo, y debe ser sorpresivo (que la gente no se entere que la están controlando).

c) De acuerdo con el tiempo en que se practica, el control puede ser:

- **Permanente:** el control se ejerce durante todo el tiempo que dura la actividad de un servicio.

- **Intermitente:** el control se ejerce a intervalos regulares, por ejemplo, cada 4 horas, e irregulares, por ejemplo, cada semana, luego cada mes, etc.

Ventajas del control

Las ventajas del control son las siguientes:

1. Permite descubrir y localizar las fallas y los errores a tiempo.
2. Revela los más mínimos incidentes o irregularidades ocurridos en el desarrollo de una actividad.
3. Permite establecer el grado de responsabilidad del personal culpable de la falla o de la irregularidad.
4. Ayuda a mantener la disciplina de los subordinados, que los jefes están obligados a mantener.
5. Permite a los jefes inmediatos estar al tanto de las actividades que desarrollan sus subalternos y les ayuda a ejercer las influencias necesarias sobre el personal para aumentar el rendimiento (Koontz, 1994).

Requisitos que debe cumplir el control

Las condiciones o requisitos que deben darse para que el control sea efectivo son:

1. Debe ser comparable: todo lo que se haga dentro del organismo, debe ser susceptible de compararse con algún modelo, patrón estándar.
2. Debe ser oportuno: el control debe aplicarse en el momento en que más se necesita información, para efectos de corregir las fallas en el momento oportuno.
3. Debe ser frecuente: El control, una vez implantado, se debe estar ejerciendo siempre o, al menos, con frecuencia.
4. Debe ser independiente: no debe existir relación de dependencia entre las personas que realizan el control y las que van a ser controladas.
5. Debe ser costo compatible: el costo debe establecer un sistema de control que no debe ser mayor al costo de la materia o el área que se va a controlar. Debe existir cierto equilibrio en el costo de ambos sistemas.

CAPITULO III

MATERIALES Y MÉTODOS

3.1 LOCALIZACIÓN

El desarrollo de la presente tesis se llevó a cabo en la ciudad de Puno, Gobierno Regional de Puno, Proyecto de Nutrición Infantil.

3.2 POBLACIÓN

Nuestra población estará integrada por los administradores que están a cargo del sistema de control de personal y tareaje conformado por 03 personas autorizadas por el Proyecto de Nutrición Infantil del Gobierno Regional Puno.

MUESTRA

Para determinar la muestra Se utilizó el diseño de muestreo no probabilístico a criterio del investigador. En ese sentido la muestra será de 03 administradores del Proyecto (n=3).

3.3 MÉTODO DE RECOLECCIÓN DE LA INFORMACIÓN

De acuerdo con lo descrito anteriormente, el enfoque mixto es el utilizado en esta investigación ya que cuenta con las características que se requieren para desarrollar el análisis de datos.

El enfoque cuantitativo se utilizó para la medición numérica de variables como grado de satisfacción y conocimiento del participante, utilizando la encuesta como técnica de recolección de datos.

El enfoque cualitativo se realizó mediante la observación, a fin de corroborar con la información extendida por el personal de la institución y verificar la autenticidad de la información obtenida mediante las entrevistas realizadas.

La Encuesta

Se realizarán preguntas abiertas y cerradas, utilizando listas de chequeo en el área de administración.

Las encuestas se utilizan para obtener información en forma escrita, a través de preguntas que se proponen. Quien responde es el responsable de la oficina de administración, el cual es el interesado. Este es el usuario potencial del sistema propuesto o será el afectado por la aplicación propuesta.

3.4 METODOLOGÍA PARA EL DESARROLLO DEL SISTEMA

3.4.1 MÉTODO DEL CICLO DE VIDA PARA EL DESARROLLO DE SISTEMAS

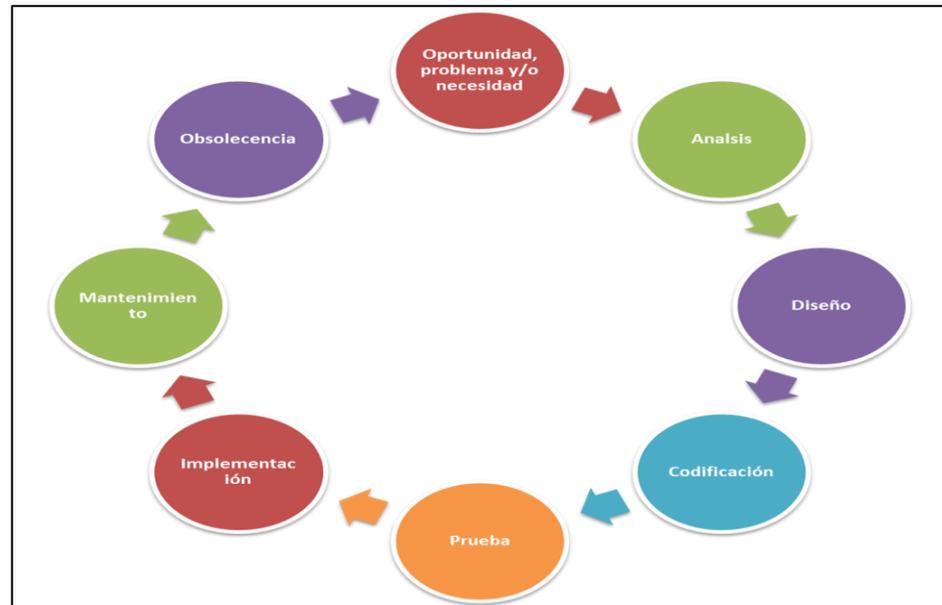
CICLO DE VIDA CLÁSICO

El software como tal sigue un proceso de desarrollo en el cual se atraviesa por diferentes fases que conformaran lo que conocemos como el ciclo de vida clásico.

Este ciclo de vida clásico recibe diferentes nombres entre ellas “ciclo de vida básico”, “modelo en cascada” o “modelo lineal secuencial”, estos nombres nos dicen que se trata de un modelo de desarrollo de software en el que tenemos que seguir una serie de etapas ordenadas de forma sistemática con el fin de generar un software.

Típicamente estos modelos de desarrollo se enfocan al software empresarial y la mayoría de los autores hacen referencia a aquel software que se desarrolla a la medida de la organización, pero esto no quiere decir que el resto del software no deba seguir las mismas etapas o fases, así que por esta ocasión trataremos de hacerlo de carácter general.

A continuación se puede apreciar un esquema con el ciclo de vida clásico de los sistemas:

Figura N° 8: Ciclo de Vida Clásico de los Sistemas

En este caso, quise mostrar el proceso desde el origen de todo el proceso, es decir, el origen del software, que como cualquier invención u obra humana surge de una necesidad, oportunidad o problema.

Sin importar cuál sea el tipo de software del que hablemos, todos surgen de algunas de esas tres simples palabras, , si tengo la necesidad de eficiente el control de calificaciones o la necesidad de conseguir un buen lugar donde comer puedo desarrollar un software que cubra cualquiera de esas necesidades, ya sean propias o de un tercero, entonces estoy desarrollando para cubrir una necesidad; si tengo el problema de que existen muchos faltantes en el almacén o que soy muy olvidadizo con los cumpleaños puedo desarrollar un software que me ayude a controlar el inventario del almacén o me ayude a recordar las

fechas de cumpleaños de las personas que conozco, entonces estoy desarrollando para solucionar un problema; y si en algún momento veo que no existe un software que cubra las necesidades de algún sector de la sociedad, puedo desarrollar un software que cubra esa necesidad, entonces estoy desarrollando porque he sabido detectar una buena oportunidad. Recordemos que todo problema o necesidad es una oportunidad.

Bien una vez detectada la necesidad el problema o la necesidad podemos comenzar el proceso:

ANÁLISIS

Esta fase realizamos una investigación a conciencia para tratar de conocer todos los requisitos e implicaciones del proyecto que estamos por abordar. Es de vital importancia que reunamos toda la información pertinente al software a desarrollar y que comprendamos la naturaleza del problema o necesidad que tratamos de cubrir. No podemos elaborar un software que solucione un problema que no comprendemos o no sabemos cómo se debe solucionar. Recordemos que las computadoras son máquinas capaces de resolver problemas muy complejos, pero que somos nosotros a través del software los que decimos como solucionar esos problemas.

Cuando se trata de un software empresarial, esta fase se enfoca en conocer no solo los requisitos o necesidades de la organización, sino también en conocer el funcionamiento de la organización,

para crear un software que responda no solo a las necesidades, sino también a la estructura y funciones de la institución.

DISEÑO

El diseño del software es la fase en donde modelamos la estructura y apariencia del software una vez que ya conocemos la naturaleza del problema y hemos determinado cual será la solución más óptima. El diseño del software se enfoca en modelar la base de datos, la arquitectura del software, la interfaz y los algoritmos o procedimientos del mismo.

En el proceso del diseño debemos construir la solución que será el software basándonos en los requerimientos obtenidos del análisis, en este sentido el diseño es el mapa o el plano que seguirá el programador para construir el software.

CODIFICACIÓN

En esta fase tomamos todo lo que se plasmó en el diseño y lo traducimos en el conjunto de órdenes para computadora que es el software.

PRUEBAS

En esta fase tomamos el software que se produjo en la codificación y probamos todas las partes y procesos del mismo, para asegurarnos que cumple con los requisitos que se habían especificado.

IMPLEMENTACIÓN

Esta es la fase en que ponemos el software en funcionamiento en el mundo real, o dentro de la organización para la que fue desarrollado. En esta fase se realizan todos los preparativos necesarios para asegurar que la inclusión del software dentro de la organización se realizara sin contratiempos y produciendo la menor cantidad de inconvenientes posible.

MANTENIMIENTO

Como sabemos las organizaciones no permanecen igual, cambian a lo largo del tiempo, así también los gustos y necesidades de las personas cambian, entonces el software necesita ser modificado para que se adapte a esos cambios, y es por ello que surge lo que en el software general las famosas actualizaciones.

OBSOLESCENCIA

Si bien es cierto que el mantenimiento hace el software se adapte a los cambios del entorno, este mantenimiento no es eterno, llega un punto en el que ya no es posible seguir haciendo modificaciones al sistema, en ese momento el software se vuelve obsoleto, ya sea por la tecnología que se usó en su desarrollo o por que no fue diseñado para la cantidad de operaciones que se realizan hoy en día, sea cual fuere la razón una vez que el software es obsoleto es tiempo de crear una nueva versión del software y es cuando volvemos a encontrar nuestra necesidad, oportunidad o problema.

3.5 MATERIAL EXPERIMENTAL

Hardware

Para el desarrollo del presente trabajo de investigación se ha utilizado

Hardware con las siguientes características:

- ✓ 01 Computadora Intel Core i3
- ✓ 4 Gb de memoria RAM.
- ✓ HD de 500 Gb.
- ✓ 01 Monitor LCD – Colores 17”
- ✓ 01 Impresora EPSON L 355.
- ✓ Memoria extraíble (Flash memory) Kingston de 4 Gb.

Software

- ✓ Lenguaje de programación VISUAL BASIC.
- ✓ Gestor de Bases de Datos SQL SERVER.
- ✓ Gestor de reportes CRYSTAL REPORT
- ✓ Microsoft Office 2013.
- ✓ Google Chrome.
- ✓ Como Sistema Operativo la plataforma Windows SEVEN.

3.6 REQUERIMIENTOS MÍNIMOS DEL SISTEMA

El programa desarrollado, por sus características tiene los siguientes requerimientos para su funcionamiento:

Para el terminal (Cliente)

- ✓ Microprocesador PENTIUM IV o superior.
- ✓ Pantalla a color (Preferible con resolución mayor a 16 bits).
- ✓ Sistema operativo: WINDOWS 98 ó superior.
- ✓ Mouse y teclado Estándar.
- ✓ HD. con un espacio libre superior a 300 Mb.
- ✓ Memoria RAM de 256 Mb. o superior.
- ✓ Tarjeta de Red.

CAPITULO IV

RESULTADOS Y DISCUSIONES

4.1 ANÁLISIS DEL SISTEMA

Para poder desarrollar el sistema de control de personal y tareaje, se realizó una secuencia de procedimientos que nos permitieron realizar de forma ordenada las diferentes etapas de la ejecución, de acuerdo a la metodología de desarrollo de software y el lenguaje Unificado de Modelado (UML).

Como primer paso, se analizó las especificaciones de los requisitos de software, que tiene como objetivo el desarrollo del “SISTEMA DE CONTROL DE PERSONAL Y TAREAJE PARA EL PROYECTO DE NUTRICIÓN INFANTIL DEL GOBIERNO REGIONAL PUNO 2015”

4.1.1. ANÁLISIS DE LOS REQUISITOS DE SOFTWARE

Investigación Preliminar: Se muestran los formatos utilizados para la obtención y recopilación de información.

Identificación de los requerimientos: Se establecen los diagramas que fueron utilizados como:

Figura N° 9: Diagrama de Contexto del Sistema



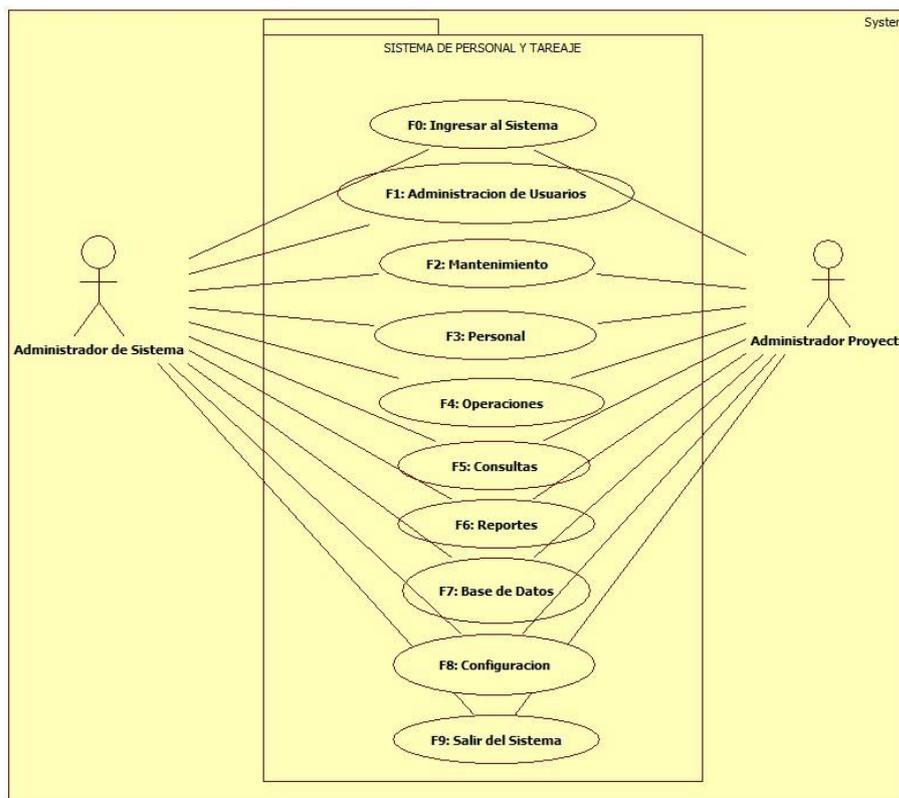
DIAGRAMAS DE CASO DE USO

Los diagramas de casos de uso representan la forma en como un administrador (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan.

Un diagrama de casos de uso consta de los siguientes elementos:

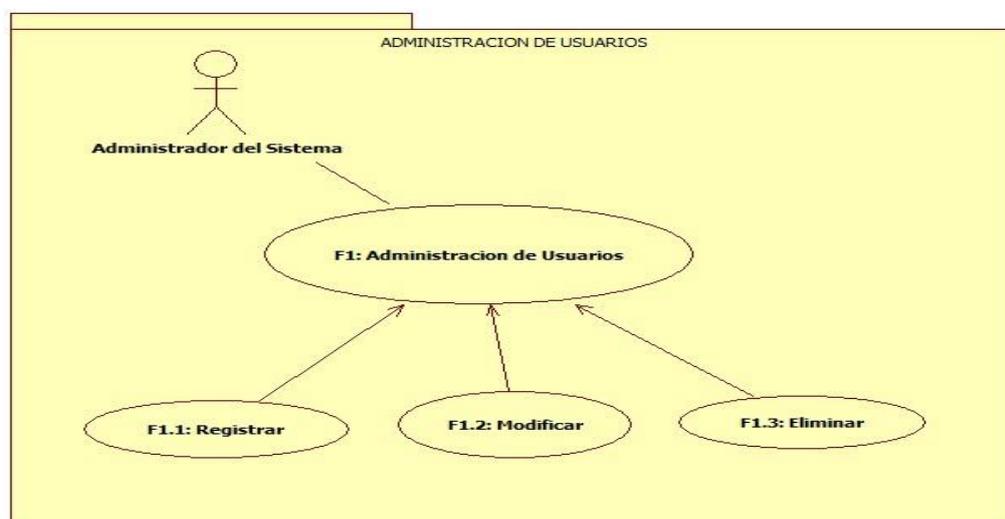
- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

Figura N° 10: Diagrama de Casos de Uso Sistema de Pernal y Tareaje



Además podemos notar que la administración de Usuarios puede hacerse mediante una acción (Registrar, Modificar o Eliminar).

Figura N° 11: Diagrama de clase de uso - Administración de Usuarios



Otro aspecto es la impresión de reportes, que puede ser realizada después de cada operación de algún trabajador por un administrado de proyecto o bien puede ser realizada a petición de un superior.

Figura N° 12: Diagrama de casos de uso - Reportes del Sistema

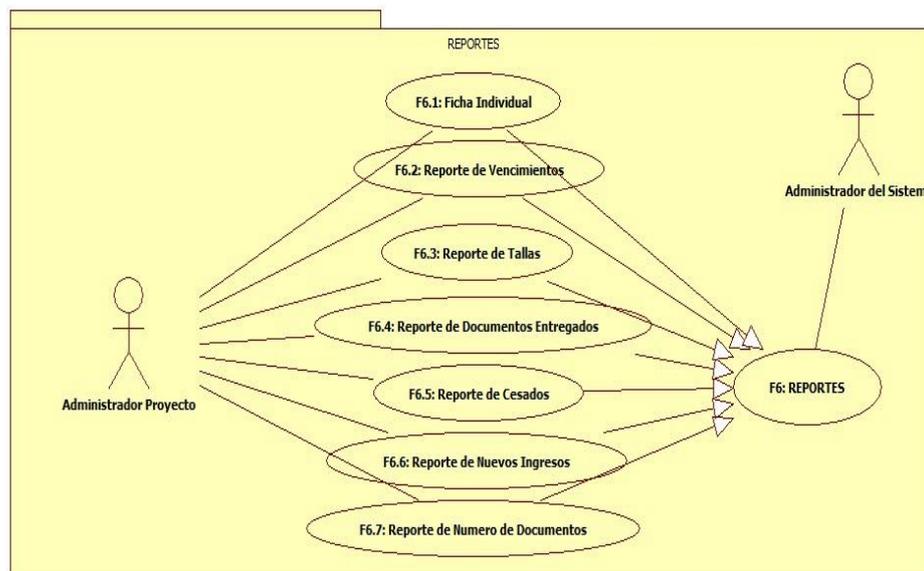
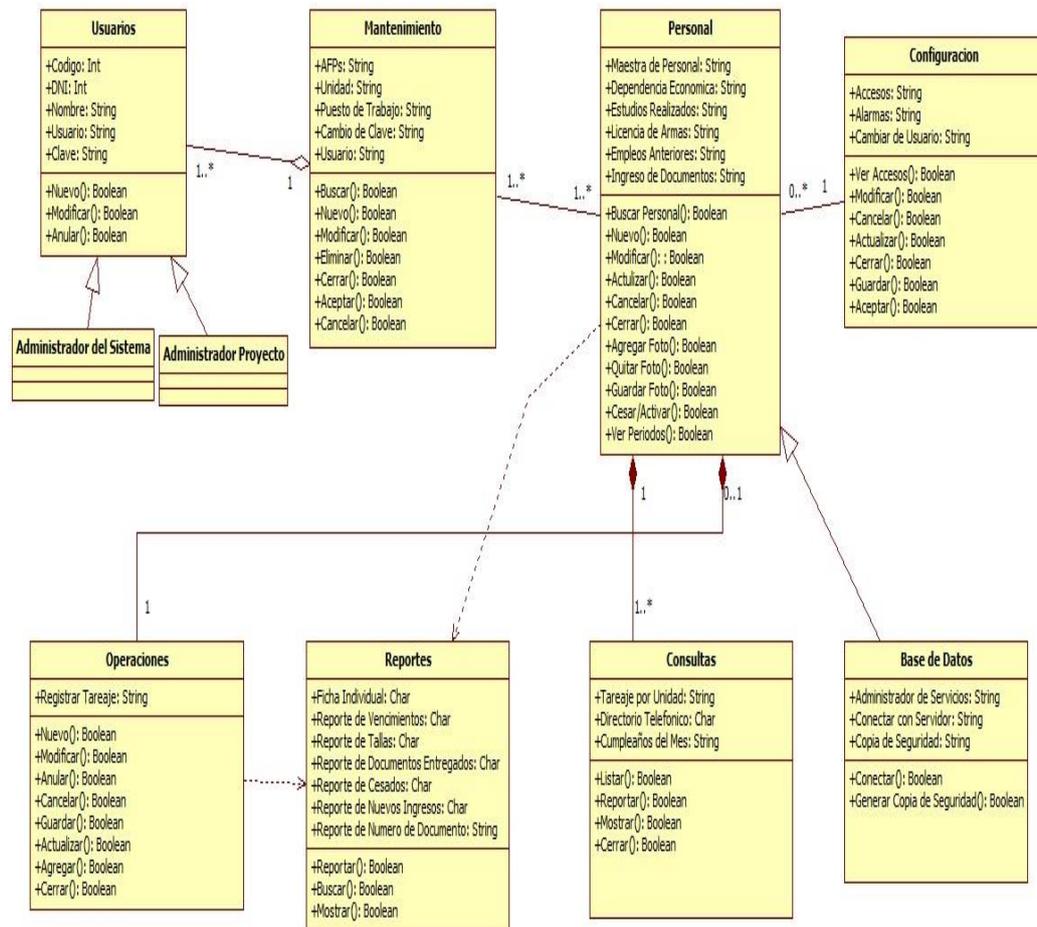


DIAGRAMA DE CLASES

En el Diagrama de Clases, Usuarios es una entidad que posee siete tipos (Mantenimiento, Personal, Configuración, Operaciones, Reportes, Consultas y Base de Datos), Estos tipos pueden ser creados, modificados, eliminados y buscados por el usuario tipo administrador. Las asistencias permiten la generación de informes administrativos tales como el tareaje, planilla de pago, etc. Todas estas entidades pueden ser creadas, modificadas, eliminadas y buscadas por los usuarios que poseen el perfil correspondiente de acceso

Figura N° 13: Diagrama de clases del sistema de personal y tareaje



Estudio de Factibilidad

Después de haber definido la problemática y haber establecido las causas que requieren de la implementación del nuevo sistema, es conveniente realizar el Estudio de factibilidad para así determinar la infraestructura tecnológica y la capacidad técnica que implica la instalación del sistema en cuestión, además de los costos-beneficios, la aceptación o rechazo que la propuesta genere a la institución.

Cabe mencionar que éste análisis nos permitirá determinar las posibilidades de diseñar el sistema propuesto, los aspectos que fueron

tomados en cuenta para su estudio, se clasificaron en tres áreas: Factibilidad Técnica, Factibilidad Económica, Factibilidad Operativa, que a continuación se describen.

Factibilidad Técnica

Para poder desarrollar el sistema de control de personal y tareaje, son necesarios los recursos tecnológicos, por tal motivo se llevó a cabo una evaluación de la tecnología con la que cuenta actualmente el Proyecto de Nutrición Infantil, se recopiló la información de éstos recursos y la posibilidad que existe de utilizarlos durante el desarrollo o quizá implantación del sistema mencionado.

La evaluación de dichos recursos tecnológicos, se realizó en base al enfoque de hardware y software. Hablando del hardware con el que cuentan, los equipos de cómputo cumplen con los requisitos básicos que se establecen, como lo es el servicio de internet inalámbrico.

En cuestión del software, el equipo que se va a adquirir, debe contar con el sistema operativo de Windows 7, ya que el personal está familiarizado con éste Sistema Operativo, al venir ya incluido en la compra de la computadora no generará un costo adicional. A continuación se muestra en la siguiente tabla las características del hardware y software.

Factibilidad Económica

Se presenta a continuación el estudio que dio como resultado la factibilidad económica del sistema donde se determinaron los recursos económicos para implementar el sistema de control de asistencia.

Costos Generales

Son todos aquellos gastos realizados como la compra de programas utilizados. Por consiguiente se muestra una tabla donde se indican dichos gastos.

Tabla 1: Costos generales

Nombre	Cantidad	Costo
SQL Server	1	\$0
Visual Basic	1	\$0

Costo de Personal

Se incluyen los gastos generados por el recurso humano que se necesita para el desarrollo del sistema únicamente, donde bajo su responsabilidad estará el contar con el personal que se encargue de operar y del funcionamiento del sistema.

Costos operativos durante el desarrollo

Estos costos se refieren a aquellos necesarios para la operatividad de las actividades durante el periodo en el que se realizara el proyecto, por lo que se suman al costo del desarrollo del mismo y al pago de servicios requeridos para laborar.

Factibilidad Operativa

Aquí se define si el sistema propuesto se pondrá en marcha, tomando en cuenta las ventajas y beneficios que traerá al personal y a la institución, además de considerar que el Proyecto de Nutrición Infantil tenga la capacidad de mantener el sistema.

La necesidad de implementar el sistema en la institución, lleva automáticamente a la aceptación de la propuesta de manera sencilla y rápida, ya que se cubre completamente con los requisitos y necesidades del usuario, y al mismo tiempo se proporcionará información exacta y oportuna, por lo que el sistema es factible al ser operado.

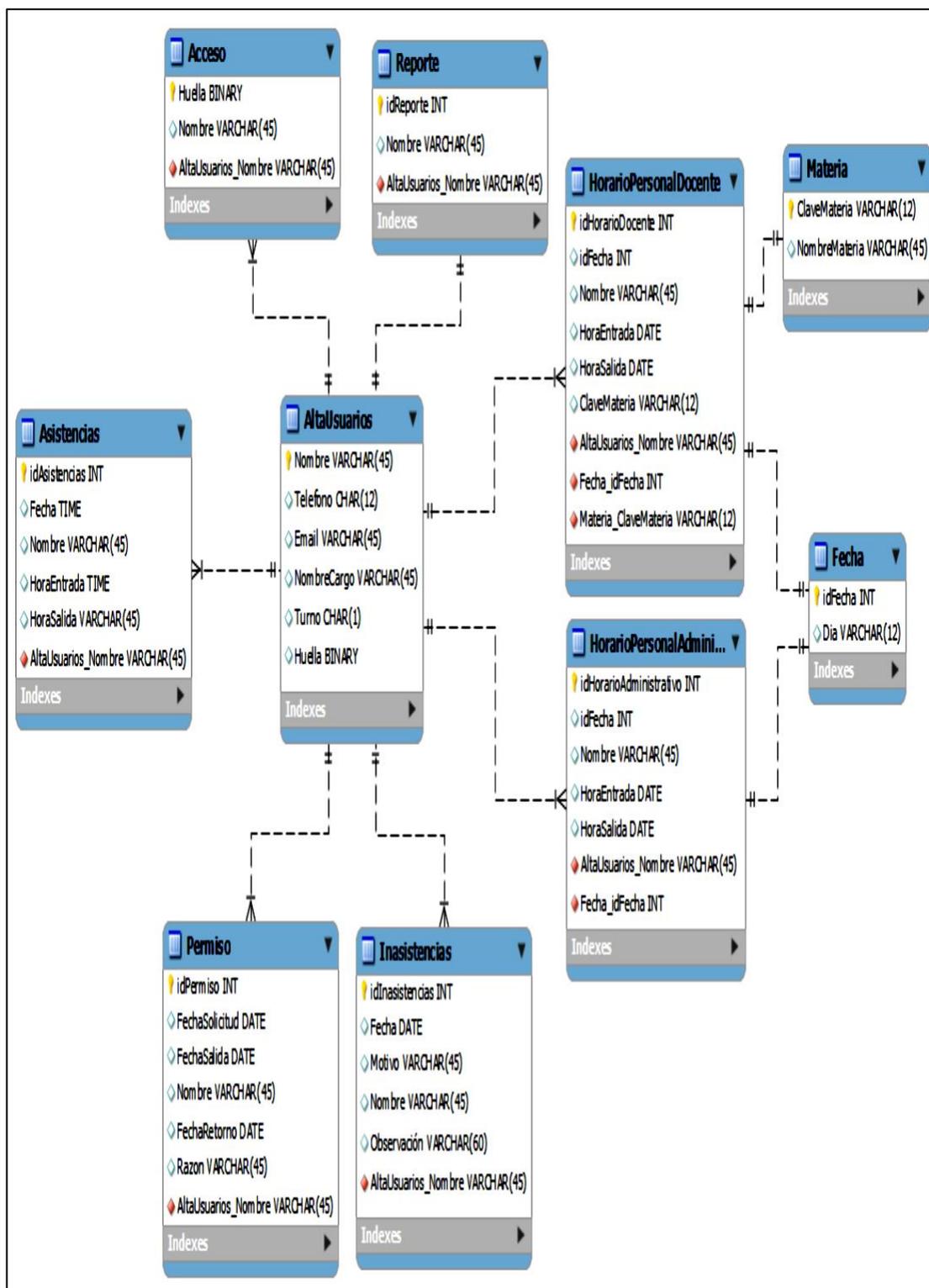
Al implantar el Sistema de Control de Personal Y Tareaje se tendrán en cuenta las siguientes restricciones:

- Los jefes de administración serán quien reciba la información de las horas laboradas de todo el personal en cada día, semana, quincena y mes.
- Quienes tendrán acceso a la manipulación o modificaciones de la base de datos únicamente serán: los administradores del Proyecto de Nutrición.

4.1.2 DISEÑO

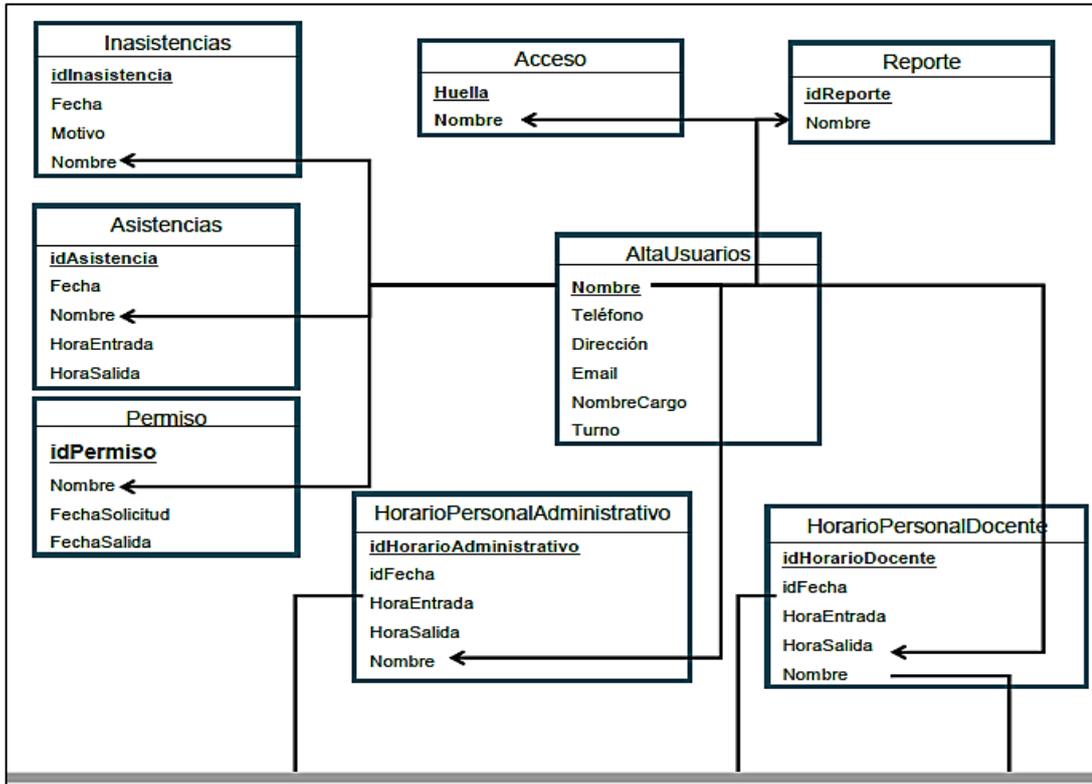
Se muestran los diagramas del diseño de cómo se encuentra conformado éste sistema, los cuales se presentan a continuación:

Figura N° 14: Diagrama del Modelo Entidad - Relación



Modelo Relacional

Figura N° 15: Diagrama del Modelo Relacional



4.1.3 REQUERIMIENTOS FUNCIONALES

Ingresar al sistema: Permite al usuario autenticarse en el sistema para utilizar los servicios del sistema.

Registrar Tareaje: Permite al usuario registrar las operaciones de los colaboradores del Proyecto de Nutrición Infantil.

Consultar: Permite al usuario realizar Consultas de la Información de los colaboradores con su respectiva asistencias y Tareajes.

Registrar información del Personal: Permite al usuario registrar datos del personal, para ello debe registrar mínimamente el N° de DNI, nombres, apellidos, lugar de trabajo, etc.

Generar Reportes: Permite al usuario realizar Consultas de la base de datos, logrando una lista con el personal colaborador por distrito, provincias y proyecto permitiendo imprimirlas.

Imprimir Tareaje: permite al usuario imprimir la el tareaje en consulta.

4.1.4 REQUERIMIENTOS NO FUNCIONALES

Existen múltiples categorías para clasificar los requerimientos no funcionales, siendo los siguientes aspectos más representativos, aunque no limitan a la definición de otros. La aplicación de la norma ISO 9126, también sirvió de aporte para la validación de los requerimientos no funcionales.

Usabilidad: Se comprobó la usabilidad de los componentes del sistema de control de personal y tareaje, aplicando los requerimientos funcionales de cada módulo, asegurando así los requerimientos no funcionales., con la aplicación de los instrumentos de evaluación ISO – 9126 para la calidad del sistema.

Confiabilidad: Se verifico la confiabilidad del sistema, corroborando la veracidad de la información en un conjunto de condiciones previamente establecidas, durante un periodo de tiempo de ejecución determinado.

Seguridad: La base de datos de encuentra conectada al servidor del Gobierno Regional donde se realizan copias de seguridad periódicamente (diario), la misma que permite la aseguración de la base de datos almacenada.

4.1.5 GENERACIÓN DE CÓDIGO

La codificación consiste en traducir el Diseño en instrucciones, que la computadora pueda interpretar.

Durante esta etapa se realizó las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en las fases de la metodología. Esta tarea la realizo el programador, siguiendo por completo los lineamientos impuestos en el diseño y en consideración siempre a los requisitos funcionales y no funcionales (ERS).

Para la codificación del sistema se optó por la plataforma Visual Basic, que es un lenguaje de programación potente, es independiente de una plataforma en específico, es decir, que cualquier programa, sistema o software creado en Visual Basic, puede funcionar en cualquier ordenador, ya sea manejando un Sistema Operativo Linux, Windows, Apple, entre otras. Visual Basic es un lenguaje de programación de propósito general, es concurrente y orientado a objetos.

4.1.6 PRUEBAS E IMPLANTACIÓN

Pruebas al Sistema.- Se llevaron a cabo algunas pruebas, las cuales se describen a continuación:

✓ **Pruebas unitarias**

En esta prueba se verificó que el código fuente no tuviera errores sintácticos o semánticos, para esto se checo el código línea por línea, de esta manera se evitó la existencia de algún error de sintaxis en alguna de las instrucciones.

✓ **Pruebas integrales**

Fue realizada para comprobar que el programa sea capaz de funcionar interactuando con la base de datos sin afectar un valor de datos en las funciones.

Se estimaron solo dos procesos en este tipo de prueba, ya que son los que comparten recursos de memoria y que pudieron haber tenido algún problema no previsto.

✓ **Pruebas de volumen**

Las pruebas de volumen fueron necesarias para identificar la forma en que se comporta el sistema, si gradualmente la información aumenta en la base de datos.

✓ **Pruebas de usuario**

Para esto se detectó las posibles fallas que una persona pueda generar, ya que no están familiarizadas con el sistema, y la forma

en que lo manipula es distinta en cada usuario; para esto se piensa capacitar al personal para que no haya ningún problema de este tipo.

Implantación

Dentro del Plan de Capacitación que se integró, se otorgó un pequeño curso a los administrativos del Proyecto de Nutrición Infantil, los cuales podrán manejar el nuevo Sistema de Control de Personal y Tareaje implementado; consistió en la enseñanza de cómo utilizar el sistema, agregar a una persona, ingresar sus datos, dar de baja, obtener las horas laboradas y faltas de personal, etc.

4.2 DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA

El sistema de control de personal y tareaje presenta un conjunto de interfaces diseñado de acuerdo a las necesidades y tendencias de desarrollo de software a nivel visual.

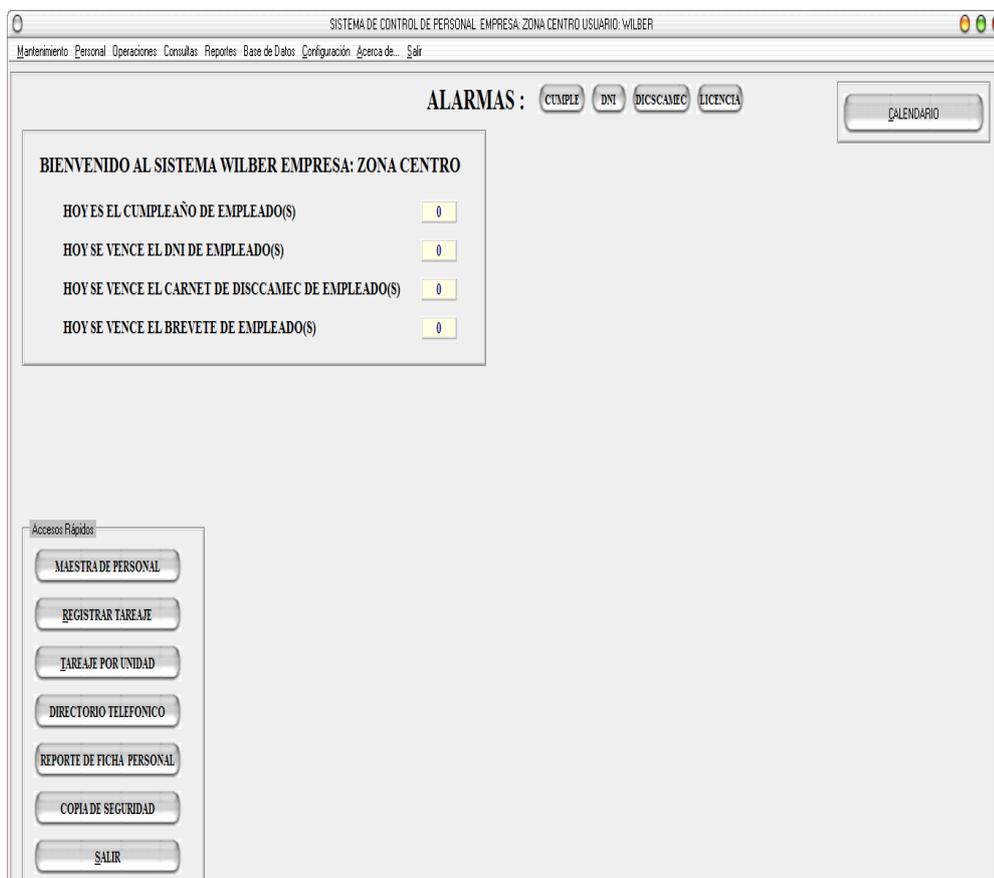
4.2.1 Pantalla de acceso al sistema

Figura N° 16: Ventana de Acceso al Sistema



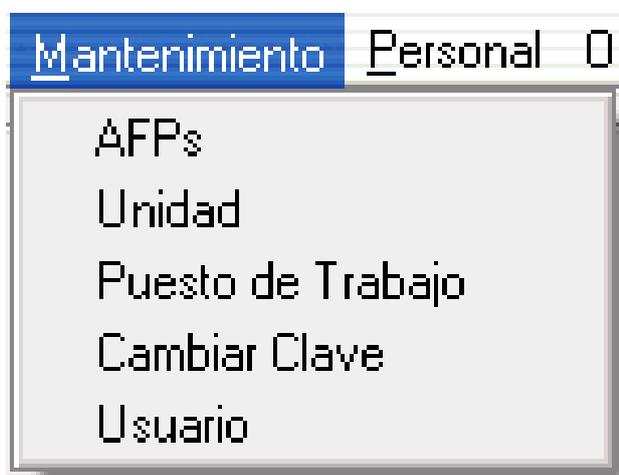
4.2.2 Ventana principal del sistema

Figura N° 17: Ventana Principal del Sistema



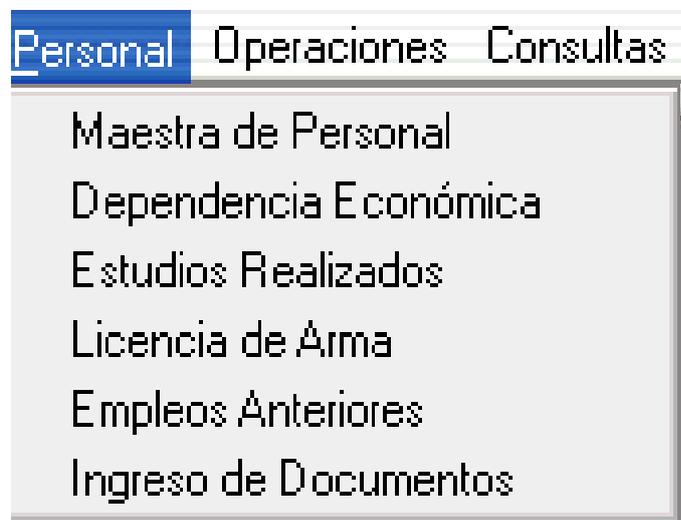
4.2.3 Modulo de Mantenimiento

Figura N° 18: Módulo de Mantenimiento del Sistema



4.2.4 Módulo registro de Personal

Figura N° 19: Modulo de Registro de Personal



4.2.5 Módulo de Operaciones

Figura N° 20: Modulo de Operaciones



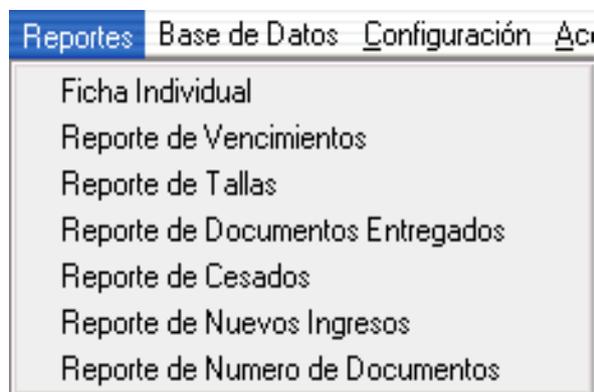
4.2.6 Módulo de Consultas

Figura N° 21: Modulo de Consultas



4.2.7 Módulo de Reportes

Figura N° 22: Modulo de Reportes



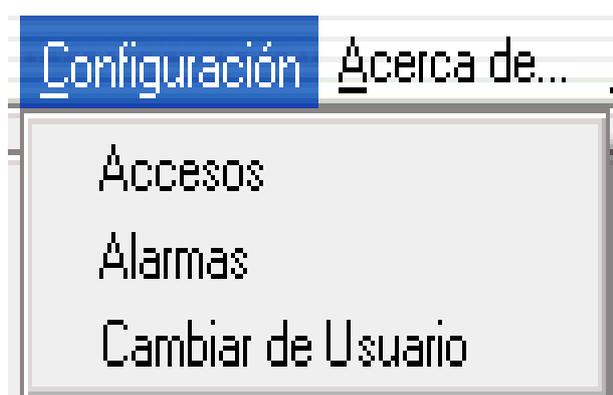
4.2.8 Módulo de Base de Datos

Figura N° 23: Modulo de Base Datos



4.2.9 Módulo de Configuración

Figura N° 24: Modulo de Configuración



4.3 PARÁMETROS DE PRUEBA DEL SISTEMA

4.3.1 Prueba de Caja Blanca

En el desarrollo del Sistema de personal y tareaje se ha considerado los aspectos necesarios en su construcción, especialmente para cada uno de los módulos que contempla, permitiendo optimizar el proceso de tareaje del Proyecto de Nutrición Infantil del Gobierno Regional de Puno.

Para lo cual se ha registrado información en el sistema y a medida que se ha ido avanzando con este proceso se fue probando los demás módulos hasta poder determinar su correcto funcionamiento y por lo que mediante esta prueba se garantiza los siguientes aspectos:

- La ejecución adecuada de cada uno de los módulos del sistema de personal y tareaje para las distintas tareas.
- Contempla procedimientos adecuados de acuerdo a la particularidad del sistema de personal y tareaje y por parte del Proyecto de Nutrición Infantil del Gobierno Regional de Puno.
- Considera el uso de sentencias de control adecuadas para el manejo de información en cada uno de los módulos.
- Existe un adecuado y medido acceso a la data del servidor.

4.3.2 Métrica de punto función

Esta medición fue tomada para evaluar la calidad que el sistema posee, para lo cual se ha considerado varios aspectos que se detalla a continuación.

a) Complejidad

Tabla 2: Valores del dominio del sistema de personal y tareaje

Parámetros de Medición	Simple		Media		Compleja		Total
	Cantidad	Peso	Cantidad	Peso	Cantidad	Peso	
Entradas	10	3	5	4	5	6	80
Salidas	20	4	10	5	5	7	165
Consultas	10	3	5	4	10	6	110
F. Lógicos o Internos	10	7	15	10	5	15	295
Interfaz	10	5	5	7	5	10	135
Total puntos de función sin ajustar (PFSA)							785

Fuente: Elaboración Propia.

b) Factores de complejidad

Para evaluar los valores de ajuste de complejidad se ha dado a cada factor valores de 0 a 5 puntos.

Tabla 3: Valores de ajuste de la complejidad del Sistema

Nº	FACTORES DE COMPLEJIDAD	VAL
01	Facilidad de corrección de módulos	5
02	Facilidad de mantenimiento	4
03	Fiabilidad del sistema	4
04	Integridad/Seguridad del sistema	4
05	Facilidad de uso para el cliente	5
06	Reutilización de código	4
07	Documentación del sistema	4
08	Copias de seguridad y de recuperación fiables	3
09	Rendimiento del sistema	4
10	Ejecución del sistema en un entorno operativo existente y fuertemente utilizado	4
11	Entrada de datos interactiva al sistema	4
12	Complejidad de entradas, salidas, archivos o las peticiones	4
13	Complejidad de procesamiento interno	5
14	Inclusión de conversión e instalación	3
15	Soportes múltiples para instalaciones en diferentes organizaciones	4
TOTAL		61

Fuente: Elaboración Propia.

Aplicando la fórmula de punto de función se tiene:

$$PF = CUENTA_TOTAL * (0.65 + 0.01 * \sum f_i)$$

Reemplazando valores:

$$PF = 785 * (0.65 + 0.01 * 61)$$

$$PF = 785 * (1.26)$$

$$PF = 989.1$$

Punto de función por mes	= 6 meses
Punto de función al mes	= $989.1/6$
	= 164.85
Costo por punto de función	= 2200
	= $2200 / 164.85$
	= 13.35
Costo del sistema	= $989.1 * 13.35$
	= 13204.49 Nuevos Soles

S/. 13, 2041.49; Es el costo estimado del Software utilizando la métrica de puntos de Función de IFPUG- FPA.

Por lo tanto: La aplicación de la métrica de punto de función nos permitieron estimar: el tiempo y el costo del Software, que resulta aproximadamente a 6 meses, con un desarrollador generando un costo estimado de 13,000 Soles.

4.3.3 PRUEBA DE SOFTWARE ESTÁNDAR ISO 9126

Después de aplicar la ficha de evaluación ISO-9126 a los 3 administradores en el proyecto de nutrición del sistema implementado, el resultado promedio se muestra en la Tabla es la siguiente:

Tabla 4: Resultados de la evaluación de la calidad del producto de software del Sistema de Personal y Tareaje según el estándar ISO – 9126

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
1. FUNCIONALIDAD					
Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.					X
Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.					X
Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas especificados				X	
Seguridad: referido a la capacidad del producto software para proteger la información y los datos				X	
Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad					X
2. FIABILIDAD					
Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software.				X	
Tolerancia a fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.				X	
Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.			X		
Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.				X	
3. USABILIDAD					
Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.					X
Facilidad de aprendizaje: la capacidad del producto software para permitir al usuario aprender su aplicación.					X
Atracción: la capacidad del producto software para atraer al usuario.				X	
Conformidad: la capacidad del producto software para Adaptarse convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.				X	
Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle.					X

INDICADORES	PUNTUACIÓN				
	1	2	3	4	5
4. EFICIENCIA					
Comportamiento temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.					X
Utilización de recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.					X
Conformidad: la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.					X
5. MANTENIBILIDAD					
Analizabilidad: Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.					X
Cambiabilidad: Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación					X
Estabilidad: Capacidad del producto software de evitar los efectos inesperados de las modificaciones.				X	
Facilidad de prueba: Capacidad del producto software de permitir validar las partes modificadas.					X
Conformidad: Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.					X
6. PORTABILIDAD					
Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.				X	
Facilidad de instalación: la capacidad del producto software para ser instalado en un ambiente determinado					X
Coexistencia: la capacidad del producto software para coexistir con otro software recursos.				X	
Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.					X
Conformidad: la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.				X	
SUB TOTALES			03	44	75
TOTAL	122				

Fuente: Elaboración Propia

Tabla 5: Tabla de valores de puntuación para la evaluación de la calidad del producto de software estándar ISO – 9126

Indicador Cualitativo	Valor
Deficiente	1
Malo	2
Regular	3
Bueno	4
Muy bueno	5

Fuente: Elaboración Propia

Tabla 6: Resultado de la Validación de la Calidad del Producto de Software del Sistema

Clasificación	Intervalo	Decisión
A) Inaceptable	[27 - 54 >	
B) Mínimamente aceptable	[54 – 81 >	
C) Aceptable	[81 – 95 >	
D) Cumple los requisitos	[95 – 122 >	*122
E) Excede los requisitos	[122 – 135]	

Fuente: Elaboración Propia

Interpretación del ISO 9126:

Se concluye que haciendo uso del ISO 9126 que mide la calidad del producto de software del Sistema de Personal y Tareaje en el Proyecto de Nutrición Infantil del Gobierno Regional Puno cumple

con los requisitos con una puntuación de 122, indicando que es conforme y el sistema es aceptable.

CONCLUSIONES

Se concluye que con el sistema de control de personal y tareaje del Proyecto de Nutrición Infantil del Gobierno Regional de Puno, es eficiente, pues se consiguió automatizar y optimizar los procesos administrativos.

También se ha realizado un correcto análisis y diseño organizacional estratégico de los procedimientos de control de asistencia de personal y tareaje con la metodología del modelo en cascada, que permitieron la cobertura de todos los requerimientos funcionales del usuario.

Las Herramientas como el UML conjuntamente con el lenguaje de programación Visual Basic y el SQL Server nos permiten diseñar y desarrollar el sistema de control de personal y tareaje, y con estas a la vez se logró cumplir con las funcionalidades de operaciones de registro, control y tareaje de manera satisfactoria.

Mediante el uso del **Estándar ISO-9126**. Se realizó la evaluación de calidad del Sistema de Control de Personal y Tareaje, obteniéndose un promedio final de **122**, confirmando que se concluye que el proyecto cumplió con los requerimientos y objetivos desde su planeación, puesto que se lograron cumplir con los objetivos específicos.

Con esta investigación se ha conseguido implementar un sistema que permita generar el reporte de las horas trabajadas por el personal y su respectivo tareaje en la Institución, con el cual se ha logrado que haya aumentado el sentido de responsabilidad y puntualidad de cada uno de los colaboradores del proyecto.

RECOMENDACIONES Y SUGERENCIAS

Se sugiere que las personas encargadas de administrar el sistema, tengan conocimientos básicos del manejo, es decir estar capacitado en la manipulación de la información registrada en el sistema de control de personal y tareaje para un control eficaz de la información y datos reales de sus empleados.

Que la información del sistema sea manipulada únicamente por los administradores de los proyectos los cuales son los únicos responsables de la información que se ingresa en el sistema para evitar cualquier alteración en los datos registrados.

El presente sistema podría adaptarse a necesidades similares en Escuelas y Colegios que no dispongan de un mecanismo de control de asistencia.

Se recomienda a la Universidad que durante el periodo de estudio se envíen mayor cantidad de proyectos prácticos, es decir trabajos que puedan ser implementados en distintos lugares, ya que esto contribuirá mucho en el desarrollo de futuros proyectos de tesis.

Es importante que el proyecto forme parte de futuras prácticas supervisadas para ampliar aún más su contenido, enriqueciéndolo con diferentes módulos de beneficio para la Facultad de Ingeniería.

BIBLIOGRAFÍA

Amaya, J. (2012). *Sistemas de Información Gerenciales*.

Bustamante, C. (2001). *Desarrollo de Sistemas Cliente Servidor*. Lima, Perú.

Campos, V. (2010). *Automatización de procesos de ingreso y control de asistencia de docentes para la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, (Universidad de San Carlos de Guatemala), 1–86*.

Cardenas, L. (2010). *Curso De Javascript*.

Correa, A. (1994). *Aplique Programación en Visual Basic 6.0, Universidad Nacional Federico Villareal. 1994 – 2000. Lima, Perú*.

Cota, A. (1994). *Ingeniería de Software. Soluciones Avanzadas – Lima, Perú*.

Eguiluz, J. (2010). *Introducción a AJAX*.

García, M. (2004). *Introducción a la Informática Editorial Mc Graw Hill. Págs.2-5*.

Gilfillan, I. (2011). *La Biblia de MySQL*.

Graham, I. (1996). *Métodos Orientados a Objetos. Segunda edición Edit. Addison-Wesley Iberoamericana S.A. Estados Unidos de América*.

Grajeda, A. (2005). *Proceso de Desarrollo de Software. “V congreso de software libre”. Cochabamba, Bolivia*.

Hemeryth, C & Sanchez, G. (2013). *Implementación de un sistema de control interno operativo en los almacenes, para mejorar la gestión de inventarios de la*

constructora A&A, de la ciudad de Trujillo-2013. Universidad Privada Antenor Orrego,

Hernandez, R. (2001). *Metodología de la Investigación. Editorial McGraw Hill Mexico.*

Kendal, I K & Kendall, J. (1997). *Análisis y Diseño de Sistemas. 3ª edición, editorial Pearson educación. México.*

Long, L. (1995). *Sistemas de Información, 4ª ed.*

López, G. (2008). *Control de asistencia al personal administrativo de la UNSM utilizando biometría.*

Maza, J. (2012). *Sistema de información y control de asistencia del personal para la facultad de ingeniería, 120, 1–120.*

Pressman, R. (1997). *Ingeniería del Software, un enfoque práctico, Editorial McGraw Hill Hispanoamérica S.A. – España.*

Beltran, J. (1990). *Sistemas Expertos. Una Metodología de Programación. Editorial Macrobit, México.*

Senn, J. (1992). *Análisis y Diseño de Sistemas de Información. Segunda Edición. Editorial McGrawHill. México.*

Sigüenza, C & Illescas, X. (2011). *Análisis, diseño e implementación del sistema de control de asistencia de personal docente y administrativo de la escuela fiscal mixta Rafael Aguilar Pesantez. Sistemas CUE - Tesis Tecnología, 187.*

Sommervill, I. (2010). *Ingeniería de Software 7th edition.*

BIBLIOGRAFÍA WEB

NOGUERA, B. (2011). Qué es MySQL, Recuperado de

<http://culturacion.com/2011/10/que-es-mysql/>

ARGENTINA, MYSQL Ya desde Cero, (2011). Recuperado de

<http://www.mysqlia.com.ar/>

Funciones MYSQL, (2012). Recuperado de

<http://php.net/manual/en/ref.mysql.php>

PHP MySQL Introducción, (2012). Recuperado de

http://www.w3schools.com/php/php_mysql_intro.asp

Mejorar el cumplimiento de la base de datos y la seguridad, (2012),

Recuperado de <http://www.oracle.com/us/products/mysql/index.html>

JQUERY, Manual de JQuery, www.w3schools.com/jquery,

JQUERY, Introducción a JQuery, jquery.com

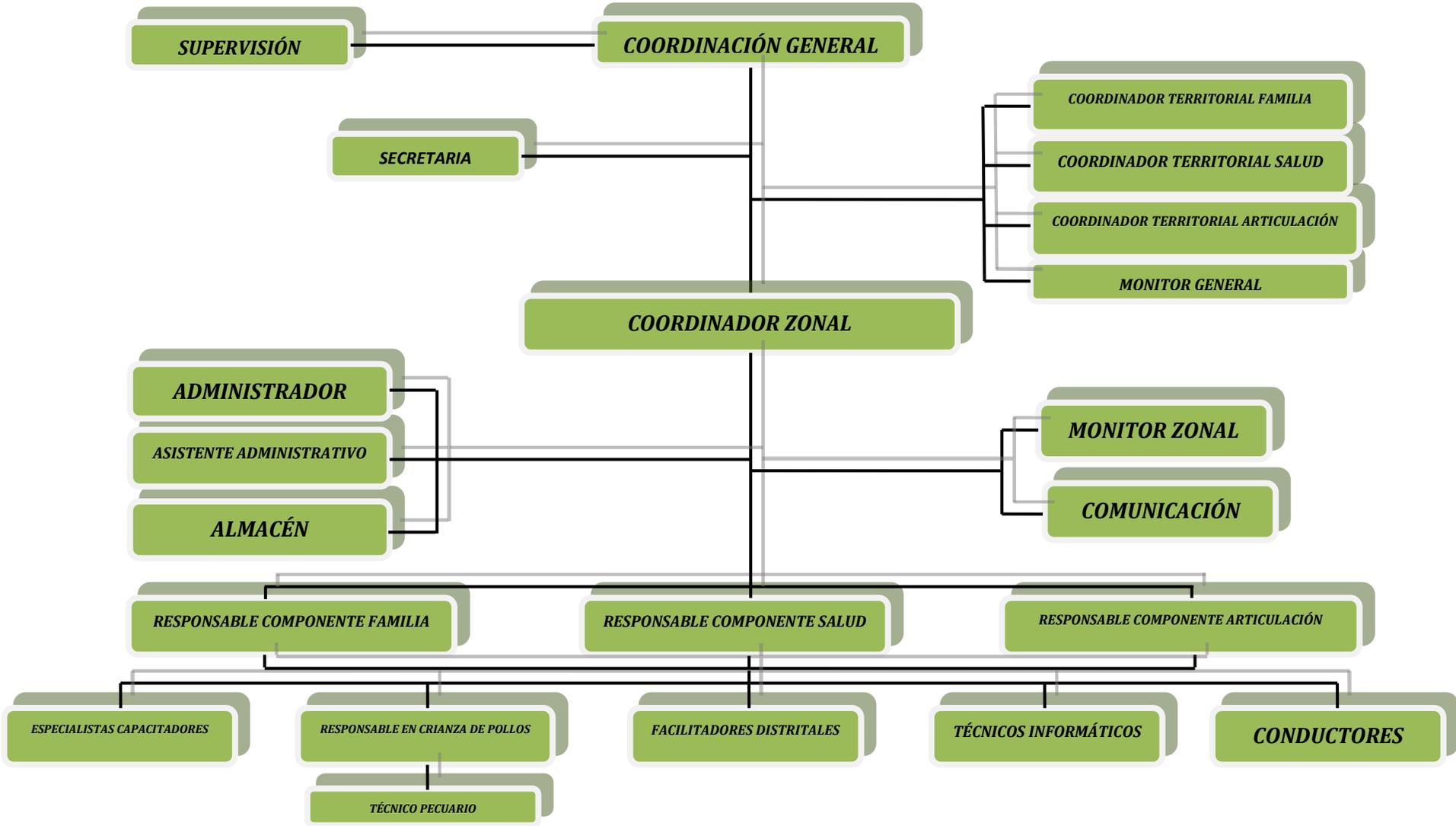
JQUERY UI, JQuery User Interface, jqueryui.com.

<http://tesis.unsm.edu.pe:8080/jspui/handle/11458/239>

http://biblioteca.usac.edu.gt/tesis/08/08_0485_CS.pdf

ANEXOS

ANEXO A: ORGANIGRAMA ESTRUCTURAL – PROYECTO DE NUTRICIÓN INFANTIL PUNO



ANEXO B: ENCUESTA SUMI

Inventario de Medida de Utilidad del Sistema (SUMI)

Instrucciones:

- 1.- La informacion que Ud. Nos brinde es persona, sincera y anonima
- 2.- Marque solo uno de las respuestas de cada pregunta, que Ud. Considere la opcion correcta.
- 3.- Debe contestar todas las preguntas.

ASPECTOS GENERALES

1. SEXO

- a) Varon
- b) Mujer

2. EDAD

- a) 21 a 25 años
- b) 26 a 30 años
- c) 31 a 35 años
- d) 36 a 40 años
- e) 41 a mas años.

3. NIVEL DE INSTRUCCIÓN COMPLETA

- a) Primaria
- b) Secundaria
- c) Universitaria
- d) Maestria
- e) Doctorado

4. EXPERIENCIA EN USO DE COMPUTADORES

- a) 1 año
- b) 2 años
- c) 4 años
- d) 5 años
- e) Mas de 5 años

5. FRECUENCIA DE ACCESO A INTERNET

- a) 1 a vez al día
- b) Varias veces al dia
- c) 1 vez a la semana

6. ULTIMO DIA DE CONEXIÓN A INTERNET

- a) La semana pasada
- b) Hace 3 días
- c) Hace 2 dias
- d) Ayer

CUESTIONARIO SUMI

Sistema de control de personal y tareaje para el
Proyecto de Nutrición Infantil del Gobierno Regional
Puno 2015

V: Control de personal

1 ¿Hay demaciados pasos necesarios para hacer un tareaje?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

2 ¿La organización de los menus paraee bastante logico?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

3 ¿Se necesita mucho tiempo para aprender las funciones del sistema?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

4 ¿Yo prefiero seguir con las funciones que mejor conosco?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

5 ¿Me siento mas seguro si uso solo unos comandos u operaciones familiar

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

6 ¿La documentacion del software es muy informativo?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

7 ¿Hay mucho que leer antes de poder empezar a utilizar este sistema?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

8 ¿Creo que este sistema es incompatible?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

9 ¿El sistema no ha hecho siempre lo que yo esperaba?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

10 ¿Me siento al mando de este sistema cuando lo estoy usando?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

11 ¿Los mensajes de error no son los adecuados?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

12 ¿Las intrucciones e indicadores son utiles?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

13 ¿A veces no se que hacer con este sistema?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

14 ¿Aprender a utilizar este sistema inicialmenete esta lleno de problemas?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

15 ¿Sigo teniendo que volver a buscar en guias?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

16 ¿Me parece que la informacion de ayuda propuesta por este software no e

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

17 ¿El sistema se presenta de una manera muy atractiva?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

18 ¿El uso de este sistema de control de personal y tareaje es una perc

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

19 ¿El sistema es realmente muy dificil?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

20 ¿Si este sistema se cuelga, se pierde la informacion?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

21 ¿Por momentos tengo que buscar mas ayuda al utilizar este sistema?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

22 ¿Este sistema en algun momento se detubo inesperadamente?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

23 ¿Me gusta el usos de este sistema de control de personal y tareaje?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

24 ¿Las pestanas de este sistema de control de personal y tareaje son

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

25 ¿Disfruto de mis sesiones con este sistema?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

ANEXO C: ENCUESTA SUMI

Inventario de Meida de Utilidad del Sistema (SUMI)

Instrucciones:

- 1.- La informacion que Ud. Nos brinde es persona, sincera y anonima
- 2.- Marque solo uno de las respuestas de cada pregunta, que Ud. Considere la opcion correcta.
- 3.- Debe contestar todas las preguntas.

ASPECTOS GENERALES

1. SEXO

- a) Varon b) Mujer

2. EDAD

- a) 21 a 25 años b) 26 a 30 años c) 31 a 35 años
d) 36 a 40 años e) 41 a mas años.

3. NIVEL DE INSTRUCCIÓN COMPLETA

- a) Primaria b) Secundaria c) Universitaria
d) Maestria e) Doctorado

4. EXPERIENCIA EN USO DE COMPUTADORES

- a) 1 año b) 2 años c) 4 años
d) 5 años e) Mas de 5 años

5. FRECUENCIA DE ACCESO A INTERNET

- a) 1 a vez al dia b) Varias veces al dia c) 1 vez a la semana

6. ULTIMO DIA DE CONEXIÓN A INTERNET

- a) La semana pasada b) Hace 3 dias c) Hace 2 dias
d) Ayer

CUESTIONARIO SUMI

Sistema de control de personal y tareaje para el
Proyecto de Nutrición Infantil del Gobierno Regional
Puno 2015

V: Gestion de informacion

1 ¿Este sistema es torpe cuando quiero hacer algo que no es estandar?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

2 ¿Este sistema me devuelve la informacion de forma rapida?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

3 ¿Este sistema de control de personal y tareaje tiene algunas caracteristicas?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

4 ¿Se demora demaciado en cargar el inicio de sesion?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

5 ¿Ingresar a este sistema es muy rapido?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

6 ¿Ingresar a este sistema es muy lento?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

7 ¿Ingresar datos a este sistema es sencillo?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

8 ¿La información que ingreso es procesada correctamente?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

9 ¿Es facil operar el sistema de control de personal y tareaje?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

10 ¿Me da lo que espero cuando hago clic en las cosas en este sistema de control?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

11 ¿Demora mucho en procesar los datos ingresados?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

12 ¿El sistema no da calculos exactos de los datos ingresados?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

13 ¿Existe ineficiencia en la carga de los datos?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

14 ¿Este sistema esta integrado con la forma de organización de la institución?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

15 ¿No me gustaria utilizar este sistema todos los dias?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

16 ¿Trabajar con este sistema es satisfactorio?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

17 ¿Yo recomendaria este sistema a mis colegas?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

18 ¿Aprendere a manejar rapidamente este sistema?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

19 ¿Este sistema de control de personal y tareaje me parece logico?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

20 ¿Este sistema de control de personal y tareaje tiene muchas cosas que mejorar?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

21 ¿Este sistema se encuentra bien estructurado?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

22 ¿Con el tiempo este sistema hara una mejor organización el la institución?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

23 ¿Este sistema no va de acuerdo a los procesos de la institución?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

24 ¿Es una perdida de tiempo usar este sistema para la organización?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5

25 ¿No satisface mis expectativas este sistema?

Nunca	Rara Vez	Algunas veces	Usualmente	Siempre
1	2	3	4	5